

The electrical conductivity structure of the Dead Sea Basin derived from 2D and 3D inversion of magnetotelluric data

Dissertation
zur Erlangung des akademischen Doktorgrades
doctor rerum naturalium (Dr. rer. nat.)
im Fachbereich Geowissenschaften
der Freien Universität Berlin

vorgelegt von

Dipl. Geophys. Naser Mohammad Meqbel
Meqbel

Berlin, Nov. 2009

Tag der Disputation: 6. November 2009

Gutachter:

PD. Dr. Oliver Ritter (1. Gutachter)

Freie Universität Berlin, GeoForschungsZentrum-Potsdam

Prof. Dr. Serge A. Shapiro (2. Gutachter)

Freie Universität Berlin

To my mother and my wife

Versicherung

Hierdurch versichere ich, dass ich meine Dissertation

“The electrical conductivity structure of the Dead Sea Basin derived from 2D and 3D inversion of magnetotelluric data”

selbständig ohne unerlaubte Hilfe angefertigt, keine anderen als die angegebenen Hilfsmittel verwendet und alle Stellen, die anderen Quellen dem Sinn nach entnommen sind, durch Angabe der Herkunft kenntlich gemacht habe. Alle wörtlich entnommenen Stellen habe ich als Zitate gekennzeichnet. Dies gilt auch für Zeichnungen, Skizzen, bildliche Darstellungen und dergleichen sowie für Quellen aus dem Internet. Die Dissertation hat in ihrer jetzigen oder einer ähnlichen Form weder ganz noch in Teilen einer in- oder ausländischen Hochschule zum Zwecke der Promotion vorgelegen.

Naser Mohammad Meqbel Meqbel, Berlin den 12. November 2009.

Abstract

This thesis has two main aspects: (i) the investigation of the deep geo-electrical structures of the Dead Sea basin (DSB) and (ii) the development of 3D inversion of magnetotelluric (MT) data using massive parallel computers.

In total, 150 magnetotelluric stations were deployed along a main profile that extends in E-W direction for ~ 110 km, crossing the southern part of the Dead Sea basin. Some areal coverage was obtained with another set of stations which were recorded along a shorter profile (~ 20 km), extending in N-S direction. The estimated transfer functions are of high to acceptable data quality. Dimensionality and directionality analyses of the measured data show that the MT stations can be subdivided in three distinct segments. MT data from eastern and western segments are consistent with 2D assumptions and indicate a stable geo-electric strike direction of $\sim N10^\circ E$. Data from the central segment appear to be more affected by underlying 3D structures.

A series of 2D conductivity models were computed for the main profile, using different combinations of data components and constrained inversion tests to analyse the stability of the various conductivity structures. The preferred 2D model consists of structures which can in general be well correlated with geological, tectonic and stratigraphical information. In particular, the transition from sedimentary formations into the crystalline basement is imaged clearly to the east and west of the Dead Sea basin. The surface traces of the eastern and western boundary faults appear in the 2D conductivity models as lateral conductivity contrasts from conductive structures in the central part to resistive structures in the eastern and western segments of the profile. The Al-Lisan salt intrusion appears as a resistive feature in the DSB. A very pronounced and robust feature in the 2D models is a sub-vertical conductor beneath the DSB which extends to depths of at least 70 km.

For a 3D interpretation of the MT data, a new parallel inversion scheme was developed in the framework of this thesis. Three-dimensional modelling of MT data requires enormous computational resources because of the huge number of data and model parameters. To overcome these difficulties the solution of the underlying differential equation systems was parallelized for different periods and electrical current system orientations (modes).

The computationally most expensive step is the linearized inversion scheme for 3D inversion, because it makes explicit use of the sensitivity matrix \mathbf{J} and cross products in form of $\mathbf{J}\mathbf{J}^T$. The dimension of this matrix depends on the number of data points and model parameters and can exceed billions of elements. Analysing the structure of this matrix revealed that it can be decomposed into blocks containing the sensitivity values corresponding to one period and one station. Distributing the computation of the blocks among several processors reduces the computation time and also the memory needed to save the sensitivity matrix. However, it is necessary to find a best compromise between communication to exchange blocks between processors and

computational performance when carrying out the cross products \mathbf{JJ}^T .

Three-dimensional inversion tests showed that the total run time can be reduced from 5 days using four processors to less than one day using 250 processors for a certain data and model parameters. At the same time, the memory requirements were reduced from 6 GB to less than 20 MB per processor for a certain data and model parameters.

The MT data collected in the Dead Sea area were inverted using the developed parallel 3D inversion scheme. The obtained 3D models show similarities and differences when compared with the 2D inversion results. There is general agreement with the conductivity structures obtained for the upper crust, i.e. the sedimentary sequences at the rift shoulders and their transition to the basement. However, the N-S extension of the Dead Sea brines, which is an obvious 3D feature, and underlying sedimentary basin are reflected in the 3D model as very conductive structures. The image of the Al-Lisan salt diapir appears more realistic in the 3D model as a confined resistive structure embedded in the high conductivity signature of the Dead Sea basin.

Zusammenfassung

Die vorliegende Dissertation betrachtet zwei Themen: (i) die Untersuchung der tiefen geoelektrischen Strukturen des Toten-Meer-Beckens (DSB) und (ii) die Parallelisierung der 3D-Inversion magnetotellurischer (MT) Daten.

Insgesamt wurden 150 Magnetotellurik-Stationen entlang eines ~ 110 km langen Hauptprofils installiert, das in Ost-West-Richtung den südlichen Teil des Toten-Meer-Beckens überquert. Im Bereich des Beckens wurde eine räumliche Abdeckung durch zusätzliche Stationen erlangt, die auf einem kürzeren (~ 20 km) Nord-Süd-Profil vermessen wurden. Die berechneten Übertragungsfunktionen besitzen eine annehmbare bis sehr gute Qualität. Dimensionalitäts- und Streichrichtungsanalysen der Messdaten ermöglichen eine Unterteilung der MT-Stationen in drei Bereiche: MT-Daten von den östlichen und westlichen Hauptprofilabschnitten zeigen ein 2D-Verhalten und weisen eine stabile geoelektrische Streichrichtung von $\sim N10^\circ E$ auf. Im mittleren Abschnitt scheinen die Daten stärker von darunter liegenden 3D Strukturen beeinflusst zu sein.

Für das Hauptprofil wurde eine Vielzahl von 2D-Leitfähigkeitsmodellen berechnet. Dabei wurden unterschiedliche Kombinationen von Datenkomponenten verwendet und Inversionstests durchgeführt, um die Stabilität der verschiedenen Leitfähigkeitsstrukturen zu untersuchen. Das bevorzugte 2D-Modell besteht aus Strukturen, die insgesamt gut mit geologischen, tektonischen und stratigraphischen Informationen korreliert werden können. Insbesondere der Übergang von sedimentären Formationen zum kristallinen Basement kann östlich und westlich des Toten-Meer-Beckens klar abgebildet werden. Unterhalb ihrer Ausbisslinien erscheinen die Eastern und Western Boundary Faults im 2D-Modell als laterale Leitfähigkeitskontraste zwischen hohen Leitfähigkeiten im zentralen Bereich und deutlich geringeren Leitfähigkeiten im östlichen bzw. westlichen Abschnitt des Profils. Die Al-Lisan Salzintrusion zeigt sich als schlecht leitfähige Struktur innerhalb des Toten-Meer-Beckens. Ein ausgeprägtes Merkmal der 2D-Modelle ist ein vertikaler Leiter unter dem Toten-Meer-Becken, der sich bis in Tiefen von mindestens 70 km erstreckt.

Zur 3D-Interpretation magnetotellurischer Daten wurde im Rahmen dieser Arbeit ein parallelisiertes Inversionsschema entwickelt. Dreidimensionale Modellierungen von MT-Daten erfordern aufgrund der großen Anzahl von Daten- und Modellparametern beträchtliche Rechnerkapazitäten. Um diese zu reduzieren, wurde die Lösung des zugrundeliegenden Systems von Differentialgleichungen über die einzelnen Perioden sowie die Richtungen (Moden) des elektrischen Stromsystems parallelisiert.

Der rechenaufwändigste Abschnitt ist das linearisierte Inversionsschema der 3D - Inversion, da es expliziten Gebrauch der Sensitivitätsmatrix J und des Kreuzprodukts der Form $\mathbf{J}\mathbf{J}^T$ macht. Die Dimensionalität der Matrix hängt von der Anzahl von Datenpunkten und Modellparametern ab und kann eine Milliarde Elemente übersteigen. Eine Analyse der Matrixstruktur zeigte, dass sie in Blöcke zerlegt werden kann, die die Sensitivitätswerte jeweils einer Periode einer Station enthalten. Eine Vertei-

lung der Berechnung der einzelnen Blöcke auf mehrere Prozessoren reduziert sowohl die Rechenzeit als auch den Speicherbedarf zur Speicherung der Sensitivitätsmatrix. Es ist jedoch erforderlich, einen guten Kompromiss zwischen dem Kommunikationsaufwand beim Austausch der einzelnen Blöcke zwischen den Prozessoren und der Geschwindigkeit bei der Berechnung des Kreuzproduktes $\mathbf{J}\mathbf{J}^T$ zu finden. Dreidimensionale Inversionstests zeigten, dass die Gesamtlaufzeit von 5 Tagen bei Verwendung von 4 Prozessoren auf weniger als einen Tag bei Verwendung von 250 Prozessoren für eine bestimmte Daten und Modelparameters verringert werden kann. Gleichzeitig konnte der Speicherbedarf von 6 GB auf 20 MB pro Prozessor reduziert werden. Die MT-Daten aus dem Gebiet um das Tote Meer wurden mittels des parallelisierten 3D-Inversionsschemas invertiert. Im Vergleich mit den 2D-Inversionsergebnissen weist das erhaltene 3D-Modell sowohl gute Übereinstimmungen als auch deutliche Unterschiede auf. Eine allgemein gute Übereinstimmung zeigt sich für die Leitfähigkeitsstrukturen der oberen Kruste, d. h. die sedimentären Abfolgen an den Riftschultern und den Übergang zum Basement. Die Nord-Süd-Ausdehnung der Toten-Meer-Solen - eine offensichtliche 3D-Struktur - und das darunter liegende sedimentäre Becken erscheinen im 3D-Modell als äußerst leitfähige Strukturen. Der Al-Lisan Salzdiapir scheint im 3D-Modell durch eine räumlich begrenzte Struktur mit hohen Widerständen innerhalb der leitfähigen Signatur des Toten-Meer-Beckens realistischer abgebildet zu sein.

Contents

Contents	vii
1 Introduction	1
2 Electromagnetic induction in the earth	5
2.1 Basic equations of electromagnetic induction in the earth	5
2.2 Induction in a 2D earth	11
2.3 Forward modeling of MT data	13
2.4 2D forward modelling with finite differences	14
2.5 Forward modelling in 3D	21
2.6 Inversion of magnetotelluric data	25
2.6.1 Linearized inversion in the model space	25
2.6.2 Sensitivity matrix computation	31
2.6.3 Solution of the normal equations	36
2.6.4 Non linear inversion using conjugate gradient (NLCG)	38
2.6.5 Linearized inversion in data space	40
3 A generic, efficient and practical parallelization method for MT forward modelling and inversion in 2D and 3D	43
3.1 Introduction	43
3.2 Introduction to parallel computing using MPI	47
3.3 Solving the MT forward modeling in parallel	57
3.4 An efficient scheme to invert MT data in 2D/3D in parallel	61
3.4.1 Computing the sensitivity matrix in parallel	63
3.4.2 Cross product $\mathbf{J}\mathbf{J}^T$ and $\mathbf{J}^T\mathbf{J}$ in parallel	65
3.4.3 Further mathematical operations with the sensitivity matrix in parallel	74
3.5 Generality and performance tests of the parallel scheme using synthetic data	76

4	Magnetotelluric measurements in the southern Dead Sea basin	91
4.1	Introduction	91
4.2	Geological and tectonic background	92
4.2.1	The structure of the Dead Sea basin and its vicinity	95
4.3	Magnetotelluric field work and data processing	99
4.4	Regional geo-electric strike direction and dimensionality analyses	106
4.5	Interpretation of the DESIRE magnetotelluric data	113
4.5.1	Evaluation of the measured data	113
4.5.2	Static shift	115
4.5.3	2D Inversion	117
4.5.4	Interpretation of the conductivity model	142
4.5.5	Three dimensional modelling and inversion of DESIRE magnetotelluric data	148
5	Conclusion and discussion	169
6	Acknowledgements	187
	Bibliography	189
	Appendices	195
A	Appendix	197

Chapter 1

Introduction

The Dead Sea basin is one of the most peculiar places on the earth, as it is (i) the deepest part on the earth (ii) it accommodates a lake with the saltiest brines. These and other curiosities do not only attract people for touristic reason, but also many geoscientists who want to understand the origin and evolution of the Dead Sea basin. The Dead Sea basin (DSB) is located along the Dead Sea transform (DST), one of the largest strike-slip faults on the earth. The DST separates the Arabian plate in the east from the African and the Sinai plates to the west. Along its ~ 1000 km length from the Red Sea extension in the south to the Taurus-Zagros continental collision zone in the north, the DST formed during its 65 million years history several compression and depression zones in form of pressure ridges and rhomb-grabens (pull apart basins), respectively. Representative for the latter is the Dead Sea basin (DSB) which is the largest along the DST and probably on the earth.

The approximately 150 km long Dead Sea basin, extends from the Wadi-Araba and Jericho strike-slip faults that delimit the southeast and northwest borders of the basin, respectively. The tectonic setting of the DSB is controlled by longitudinal strike-slip and normal faults which are the most prominent tectonic elements controlling the basin structure. Sedimentation in the DSB started in the early Miocene when uplifting which is associated with the DST and subsidence of the DSB occurred. The successive anharmonic phases of subsidence and sedimentation cause the deep topographic trough of the DSB. This makes the floor of the Dead Sea brines the deepest point on the earth. Magnetic, gravity and seismic measurements supported by geological studies inferred the thickness of the sedimentary basin fill to be ~ 10 km. Moreover, these studies inferred the presence of several salt intrusions in the DSB in form of diapirs.

Understanding the internal structure of the DSB and its tectonic history is a major step towards understanding the geodynamic process that occur at plate tectonic

boundaries. Therefore, several geophysical measurements and geological studies were carried out over the years in the Dead Sea area. However, most of these studies were confined to the western or eastern sides of the DSB due to the difficult political situation across the borders between Jordan to the east and Palestine and Israel to the west. Recently, a series of multi-disciplinary multi-national projects started in the middle east, in particular along the DST, gathering geo-scientists from the region and outside to derive an integrated picture of the structures and evolution of the DST. Most recently, the German Research Foundation (DFG) and German Research Centre for Geosciences (GFZ) founded DESIRE-project (Dead Sea Integrated Research) focuses on the southern part of the DSB. Along a ~ 300 km E-W traverse crossing the southern part of the DSB, several geophysical, geological and geo-dynamical studies were carried out. The geophysical methods applied involve seismic and magnetotelluric, both of which are capable to image on a lithospheric scale, however, two different physical properties are resolved: The elastic properties and the electric conductivity. The magnetotelluric method uses the natural variations of the electric and magnetic fields (EM) over a wide range of periods to map the distribution of the sub-surface resistivity. These natural EM fields are generated in the Earth's atmosphere mainly by lightning and interaction between the solar wind and the ionosphere. For the MT method, the horizontal components of the electric field and all three orthogonal components of the magnetic field are measured at the earth's surface. From these measurements, the ratios of electric to magnetic field components (transfer functions) can be determined to calculate the so-called impedance tensor which contains the information of the three-dimensional electrical resistivity distribution of the sub-surface. The electrical resistivity of rocks varies from highly conductive ($10^{-5} \Omega m$ for ore deposits) to highly resistive ($10^8 \Omega m$ for crystallised igneous rocks). The diffusion nature of the electromagnetic fields implies that long period fields propagate deeper in the earth; this is the so-called skin effect. The dependency on period means that transfer functions yield an apparent resistivity (ρ_a) and phase (ϕ) but not directly the "true" resistivity of the sub-surface. Like in any geophysical method, however, we are interested to obtain the spatial distribution of the "true" physical quantities of the sub-surface. Mapping from measured data to a "true" conductivity model requires an inversion. The opposite way to convert a "true" conductivity model to data is called forward modelling or simulation. The forward modelling process takes into account the physical properties of the conductivity model (resistivity distribution, periods, etc) to simulate the spatial variations of the electromagnetic field in and on the "model" earth. Whereas, the inversion process is a mathematical formulation which aims to minimise the differences between the forward modelling process (synthetic data) and the measured data to compute a conductivity model. The inversion contains the forward modelling and both processes interact in an iterative manner to obtain a conductivity model which can explain the measured data

within certain error bounds.

The conductivities model can be one dimensional (1D) in which the conductivity of the earth varies only in the vertical direction ($\sigma(z)$) or two dimensional (2D) in which the conductivity varies only in the vertical direction and one horizontal direction (i.e. $\sigma(y, z)$). In a three dimensional model (3D) which resembles the real world best, the conductivity of the earth is allowed to vary in all three dimensions ($\sigma(x, y, z)$). For 1D models there exist analytic solutions for the forward modelling problem and even for solving the inverse problem. However, due to the geological complexity of the sub-surface, analytic solutions to solve the forward modelling problem in 2D and 3D do not exist. Instead, numerical solutions are sought to approximate the spatial variation of the electromagnetic fields. For numerical solutions, the sub-surface is commonly subdivided into small cells (model parameters) and each cell is assigned to a constant conductivity. Then, the spatial variations of the electromagnetic fields are approximated by solving Maxwell's equations which describe precisely the diffusion and the attenuation of the electromagnetic field in the earth.

Solving the forward modelling problem numerically in 2D means models with several thousands of parameters. However, the number of model parameters in a 3D model may easily exceed several millions of cells and solving Maxwell's equations for each of these model parameters is impracticable on today's processors.

To overcome the difficulty in solving huge numerical problems that arise in modern natural sciences, the concept of parallel computing was introduced in the 1960ies. As the name suggests, parallel computing is based on distributing the computational tasks on several computers, ensuring that each computer solves only one task. In this case the run time required to solve a problem on only one computer could ideally be reduced by a factor equal to the number of available computers. When using the concept of parallel computing, both run-time and the memory needed to save various parameters are reduced. The latter is very important, since solving the 3D forward modeling and inverse problems require a huge amount of computer memory.

The geological complexity and the tectonic setting of the DSB and therefore the complex DESIRE-MT data represent a big challenge for forward modelling and inversion. This means a considerable effort and extreme care is necessary to interpret DESIRE-MT data in 2D which is still the most common way as it is considerably faster than 3D and can be run on a personal computer. However, a 2D interpretation of the MT data which are influenced by a 3D effect can lead to a misinterpretation. Because of that, the first challenge is to find a proper 2D conductivity model which can explain the 2D nature of the data set. This is, however, not a straightforward task and requires first data analysis to explain the 2D and 3D nature of the data set. Moreover, resolution studies must be applied for the obtained 2D conductivity models to verify that the data can be explained satisfactory. Comparing the obtained 2D conductivity model with geological cross-sections, lithological columns and another geophysical models

available along the DESIRE-MT profile, provides more confidence to the obtained 2D conductivity model and can better explain the structures of the DSB.

The main challenge is to model the DESIRE-MT data in 3D. This is because the available computer programs to solve the forward modelling and inverse problems in 3D are, to some extent, limited in their capability to deal with the huge number of model parameters. The need of a huge number of model parameters to model the DESIRE-MT data in 3D is based on facts that, i) the 3D features present in the DSB have relative small scales (salt diapirs), which requires small cells sizes for exact localisation, ii) the Dead Sea with its underlying basin (~ 150 km length, ~ 20 km width and ~ 10 km depth) must be modelled, iii) regional conductivities structures that are located in the vicinity of the study area (i.e. the Mediterranean Sea), must be considered in the 3D model and iv) MT data with six complex transfer functions (four impedance tensor and two vertical magnetic transfer functions) and 150 sites have to be explained. Combining all these features in one model results in huge number of model parameters. Even if the required memory is available to deal with such a huge model, run time of several weeks are required to solve the 3D modelling problem.

These difficulties having to model the DESIRE-MT data in 3D guide me to adopt the parallel computing concept. The parallel schemes I have developed in the framework of my PhD-thesis show their efficiency in modelling MT data in 3D. Beside the run time reduction that can be reached when applying the developed parallel schemes, memory-efficient concepts are also considered in the 3D modelling. The application of these two concepts to 3D modelling allows us to construct huge models for regional and crustal studies in MT.

Chapter 2

Electromagnetic induction in the earth

All electrical and electromagnetic (EM) methods aim to map the electric properties of the subsurface from surface measurements. In contrast to the other EM methods, in which an active source is used, the MT method uses the propagation and attenuation of natural electromagnetic fields.

Understanding the processing and the interpretation techniques used later, requires first a knowledge of the physical theories standing behind them. In general, the electromagnetic theory satisfies the description of the EM methods. This theory based on Maxwell equations to depict the propagation and diffusion of electromagnetic waves in free space as well in materials.

In the following sections, the basic concept of the MT method and the main formulations of the induction problem will be discussed from physical point of view to end up with equations used later in solving the forward modelling problem in two and three dimensions (2D and 3D).

2.1 Basic equations of electromagnetic induction in the earth

The propagation and attenuation of the electromagnetic (EM) fields is described elegantly and concisely by Maxwell's equations. These equations describe the relation between the time varying electric and magnetic fields. The first two Maxwell equations

in their differential form are:

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad (2.1)$$

$$\nabla \times \mathbf{H} = \frac{\partial \mathbf{D}}{\partial t} + \mathbf{J} \quad (2.2)$$

Here, \mathbf{E} is the electric field (V/m), \mathbf{B} is the magnetic flux density (Wb/m²), \mathbf{H} is the magnetic field (A/m), \mathbf{D} is the electric displacement current (C/m²), and \mathbf{J} is the current density (A/m²).

Equation 2.1 represents Faraday's law which indicates that the induced electric field is equal to the time rate of change of the magnetic flux. The modified (after Maxwell) Ampere's law in eq. 2.2 relates the magnetic field with the electric current density and the electric displacement current.

Two more Maxwell's equations can be derived from eqs. 2.1 and 2.2 by taking the divergence ($\nabla \cdot$) of eq. 2.1 and using the vector identity ($\nabla \cdot \nabla \times A = 0$):

$$\begin{aligned} \nabla \cdot \nabla \times \mathbf{E} &= -\nabla \cdot \frac{\partial \mathbf{B}}{\partial t} = -\frac{\partial}{\partial t}(\nabla \cdot \mathbf{B}) = 0 \\ \implies \nabla \cdot \mathbf{B} &= 0 \end{aligned} \quad (2.3)$$

Equation 2.3 states that magnetic monopoles do not exist (Gauss's law for magnetism). Similarly, by taking the divergence of eq. 2.2 we obtain:

$$\begin{aligned} \nabla \cdot \nabla \times \mathbf{H} &= \frac{\partial}{\partial t}(\nabla \cdot \mathbf{D}) + \nabla \cdot \mathbf{J} = 0 \\ \implies \nabla \cdot \mathbf{J} &= -\frac{\partial}{\partial t}(\nabla \cdot \mathbf{D}) \end{aligned} \quad (2.4)$$

Using the continuity equation, which states that the divergence of the current density is equivalent to the rate of accumulation of charge density q , in eq. 2.4 follows that:

$$\nabla \cdot \mathbf{J} = -\frac{\partial q}{\partial t} = -\frac{\partial}{\partial t}(\nabla \cdot \mathbf{D}) \quad (2.5)$$

thus,

$$\nabla \cdot \mathbf{D} = q \quad (2.6)$$

Equation 2.6 shows that the electric field is the result of the distribution of electric charge (Gauss's law for electricity).

Equations 2.1, 2.2, 2.3 and 2.6 represent the fundamental equations in electromagnetism (Ward & Hohmann Gerald W., 1987). For linear, isotropic media of electric conductivity σ , magnetic permeability μ and electric permittivity ϵ , three further relationships have been shown to hold:

$$\mathbf{B} = \mu \mathbf{H} \quad (2.7)$$

$$\mathbf{D} = \epsilon \mathbf{E} \quad (2.8)$$

$$\mathbf{J} = \sigma \mathbf{E} \quad (2.9)$$

Since the variations of the dielectric permittivity ϵ and the magnetic permeability μ for most rocks are very small in comparison to the variations of the electric conductivity σ , we can use the free-space values for both μ and ϵ and set $\mu = \mu_0 = 1.25566 \times 10^{-6}$ H/m, $\epsilon = \epsilon_0 = 8.85 \times 10^{-12}$ F/m. Using eqs. 2.7, 2.8 and 2.9, Maxwell's equations can be rewritten as:

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad (2.10)$$

$$\nabla \times \mathbf{B} = \mu_0 \epsilon \frac{\partial \mathbf{E}}{\partial t} + \mu_0 \sigma \mathbf{E} \quad (2.11)$$

$$\nabla \cdot \mathbf{E} = q/\epsilon_0 \quad (2.12)$$

$$\nabla \cdot \mathbf{B} = 0 \quad (2.13)$$

To derive the induction equation in the Earth, the following assumptions must be considered when applying the magnetotelluric technique:

1. The natural electromagnetic source field used in MT is generated by large-scale ionospheric current systems. The origin of these current system is far away from the Earth's surface so that the electromagnetic field can be treated as uniform, plane-polarized wave impinging on the Earth at near-vertical incidence (Cagniard, 1953). For plane electromagnetic waves, the electric \mathbf{E} and magnetic \mathbf{B} fields with amplitude \mathbf{E}_0 and \mathbf{B}_0 at origin and angular frequency ω have the mathematical form:

$$\mathbf{E} = \mathbf{E}_0 e^{i\omega t} \quad (2.14a)$$

$$\mathbf{H} = \mathbf{H}_0 e^{i\omega t} \quad (2.14b)$$

2. The electric displacement field within the conductive Earth is quasi-static in MT. Therefore, time-varying displacement currents are negligible compared with the time-varying conduction currents. This means that eq. 2.11 takes the form:

$$\nabla \times \mathbf{B} = \mu_0 \sigma \mathbf{E} \quad (2.15)$$

3. There is no charge accumulation within a layered earth. However, within a multi dimensional earth, charges can only accumulate at conductivity gradients. This phenomena is known as the galvanic effect (Weaver, 1994; Jiracek, 1990). The physical meaning of this phenomena can be explained as follows: Taking the divergence of eq. 2.9,

$$\nabla \cdot \mathbf{J} = \nabla \cdot (\sigma \mathbf{E}) = \sigma \nabla \cdot \mathbf{E} + (\nabla \sigma) \cdot \mathbf{E} = 0 \quad (2.16)$$

combined with eq. 2.12 gives the expression:

$$q = -\epsilon_0 (\nabla \sigma) \cdot \mathbf{E} / \sigma \quad (2.17)$$

Equation 2.17 states that a conductivity contrast between two adjacent media with σ_0 and σ_1 causes accumulation of charges at the interface between them to guarantee continuity of currents (Current conservation law). The galvanic effect is visualized in Fig. 2.1.

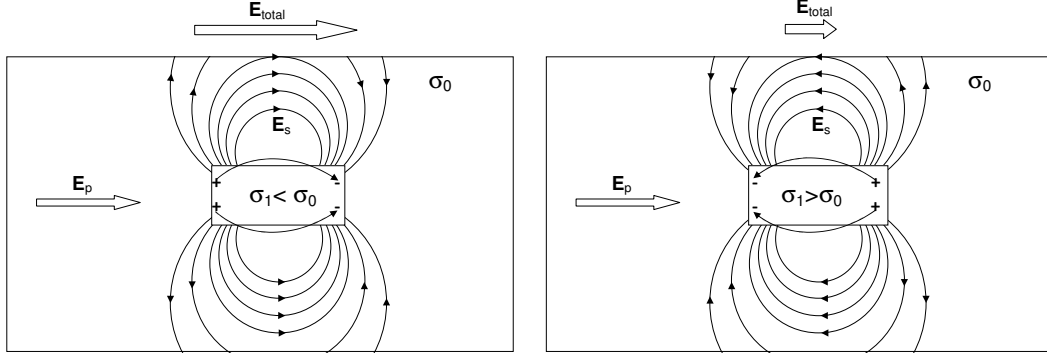


Figure 2.1: Galvanic effect. Boundary charge accumulation of the surface of (a) a resistive body embedded within conductive background (b) conductive body within resistive background. In both cases, the charges produce secondary electric field \mathbf{E}_s . Primary fields (\mathbf{E}_p) and secondary field (\mathbf{E}_s) are superimposed and amplify (a) or reduce (b) the total electric field (\mathbf{E}_{total}) (Jiracek, 1990).

Equations 2.10, 2.12 and 2.15 represent the basic equations of electromagnetic induction in a source-free medium. The fields that satisfy them are time-dependent, but they change sufficiently slowly that their spatial distribution behaves like a static field. Therefore, these fields are called quasi-static fields.

From the preceding discussion it follows that for quasi-static fields the differential equation satisfied by \mathbf{E} is given by taking the curl ($\nabla \times$) of eq. 2.10 and substituting $\nabla \times \mathbf{B}$ from eq. 2.15, namely:

$$\begin{aligned} \nabla \times (\nabla \times \mathbf{E}) &= -\frac{\partial}{\partial t}(\nabla \times \mathbf{B}) \\ &= -\mu_0 \sigma \frac{\partial \mathbf{E}}{\partial t} \end{aligned} \quad (2.18)$$

Introducing the vector Laplacian operator:

$$\nabla^2 := \nabla(\nabla \cdot) - \nabla \times (\nabla \times)$$

and substituting for $\nabla \cdot \mathbf{E}$ from eq. 2.16, we may rewrite eq. 2.18 in the alternative form:

$$\nabla^2 \mathbf{E} = \mu_0 \sigma \frac{\partial \mathbf{E}}{\partial t} - \nabla[(\nabla \sigma) \cdot \mathbf{E} / \sigma] \quad (2.19)$$

Similarly, the corresponding differential equation satisfied by \mathbf{B} can be derived by taking the curl of eq. 2.15 and substituting from eq. 2.10:

$$\begin{aligned}\nabla \times (\nabla \times \mathbf{B}) &= \mu_0 \sigma (\nabla \times \mathbf{E}) \\ &= -\mu_0 \sigma \frac{\partial \mathbf{B}}{\partial t}\end{aligned}\quad (2.20)$$

Using the vector Laplacian operator and substituting for $\nabla \cdot \mathbf{B}$ from eq. 2.13 we deduce that:

$$\nabla^2 \mathbf{B} = \mu_0 \sigma \frac{\partial \mathbf{B}}{\partial t} - [(\nabla \sigma) \times (\nabla \times \mathbf{B})/\sigma] \quad (2.21)$$

Equations 2.19 and 2.21 take the form of diffusion equation in terms of time varying electric and magnetic fields. Thus, the electromagnetic wave propagates diffusively through the earth and dissipates exponentially (Weaver, 1994).

Considering assumption number 1 and using a plane wave with harmonic time dependency term ($e^{i\omega t}$) we can evaluate the derivative of \mathbf{E} with respect to the time in eq. 2.19 to give:

$$\nabla^2 \mathbf{E} = i\omega \mu_0 \sigma \mathbf{E} - \nabla [(\nabla \sigma) \cdot \mathbf{E}/\sigma] \quad (2.22)$$

Similarly, eq. 2.21 gives:

$$\nabla^2 \mathbf{B} = i\omega \mu_0 \sigma \mathbf{B} - [(\nabla \sigma) \times (\nabla \times \mathbf{B})/\sigma] \quad (2.23)$$

Within a uniform conductor where $\nabla \sigma = 0$, eqs. 2.22 and 2.23 reduced to:

$$\nabla^2 \mathbf{E} = i\omega \mu_0 \sigma \mathbf{E} = k^2 \mathbf{E} \quad (2.24)$$

$$\nabla^2 \mathbf{B} = i\omega \mu_0 \sigma \mathbf{B} = k^2 \mathbf{B} \quad (2.25)$$

where $k = \sqrt{i\omega \mu_0 \sigma}$ is the complex wave number. In non-conducting regions, i.e. air layer, where $\sigma = 0$, we get (Weaver, 1994):

$$\nabla^2 \mathbf{E} = 0, \quad \nabla^2 \mathbf{B} = 0$$

Resolving the square root in the definition of the complex wave number yields:

$$k = \sqrt{i\omega \mu_0 \sigma} = \sqrt{i} \sqrt{\omega \mu_0 \sigma} = \frac{1+i}{\sqrt{2}} \sqrt{\omega \mu_0 \sigma} = (1+i) \sqrt{\omega \mu_0 \sigma / 2} \quad (2.26)$$

Equation 2.26 shows that the wave number has equal real and imaginary parts (Keller, 1988). The real part of eq. 2.26 is:

$$Re(k) = \frac{1}{p} \quad (2.27)$$

where $p = \sqrt{2/\omega \mu_0 \sigma}$ and known as a *skin depth*. The *skin depth* is usually used as a criterion for the penetration of the electromagnetic wave. It describes in which depth

the signal is reduced to $1/e$ of its original strength. For resistivity in Ωm and period in s , the skin depth is given in m as:

$$p \approx 500\sqrt{\rho T} \quad (2.28)$$

The magnetotelluric impedance tensor (\mathbf{Z}) describes the linear relationship between the orthogonal components of the electric and magnetic fields, computed from the ratio of the electric to the magnetic field components as:

$$Z(\omega) = \frac{E(\omega)}{B(\omega)} \quad (2.29)$$

Using all possible horizontal components for both electric and magnetic fields, the impedance can be written in a tensor form as:

$$\begin{pmatrix} E_x \\ E_y \end{pmatrix} = \begin{pmatrix} Z_{xx} & Z_{xy} \\ Z_{yx} & Z_{yy} \end{pmatrix} \cdot \begin{pmatrix} B_x \\ B_y \end{pmatrix} \quad (2.30)$$

The complex impedance tensor in eq. 2.30 can be converted into apparent resistivity and a phase, namely:

$$\rho_a^{ij} = 0.2T \left| \frac{E_i}{B_j} \right|^2 = 0.2T |Z_{ij}|^2 \quad (2.31)$$

$$\Phi^{ij} = \arctan \left(\frac{\text{Im}(Z_{ij})}{\text{Re}(Z_{ij})} \right) \quad (2.32)$$

where $i,j=x,y$ and presuming that the electric field is measured in mV/km and the magnetic field in nT , i.e. (Cagniard, 1953; Keller, 1988; Weaver, 1994). Further properties of the complex impedance tensor \mathbf{Z} are discussed in literature i.e. (Weckmann *et al.*, 2003)

Another linear relationship of the electromagnetic fields is the relation between the vertical magnetic component B_z and the horizontal magnetic field components B_x, B_y , which is written as:

$$B_z = T_{zx}B_x + T_{zy}B_y \quad (2.33)$$

where T_{zx} and T_{zy} constitute the vertical magnetic transfer functions. They describe to which extent the horizontal magnetic components are tipped into the vertical components. T_{zx} and T_{zy} are sometimes called "tipper" (Vozoff, 1972). An important quantity derived from the vertical magnetic transfer functions is the induction vector: Using Wiese-Convention (Wiese, 1962) the amplitude and the angle of the real part of the induction vector are computed as:

$$\text{amplitude} \quad \sqrt{\text{Re}(T_{zx})^2 + \text{Re}(T_{zy})^2} \quad (2.34)$$

$$\text{angle} \quad \arctan \left(\frac{\text{Re}(T_{zy})}{\text{Re}(T_{zx})} \right) \quad (2.35)$$

The imaginary part is computed accordingly. In Wiese-convention the real induction vectors tend to point away from the elongated conductors. Induction vectors are usually used to identify lateral contrasts of the conductivity in the subsurface.

2.2 Induction in a 2D earth

In 2D earth models, we assume that the conductivities of the earth vary in the vertical and one horizontal directions (Fig. 2.2), and approach 1D distribution as $y \rightarrow \pm\infty$ $\lim_{y \rightarrow \pm\infty} \sigma(z, y) = \sigma(z)$. From now on, we will assume that conductivity varies in yz-plane, and the x-axis coincides with the geo-electric strike direction, where $\nabla_x \sigma = 0$. To analyse the behavior of a plane electromagnetic wave in a 2D earth, let a linearly polarised electromagnetic wave, characterized by the wavenumber vector \mathbf{K} , be incident on the model at any angle and azimuth relative to the earth surface (Fig. 2.2a). The incident angle is left completely arbitrary (Cagniard, 1953).

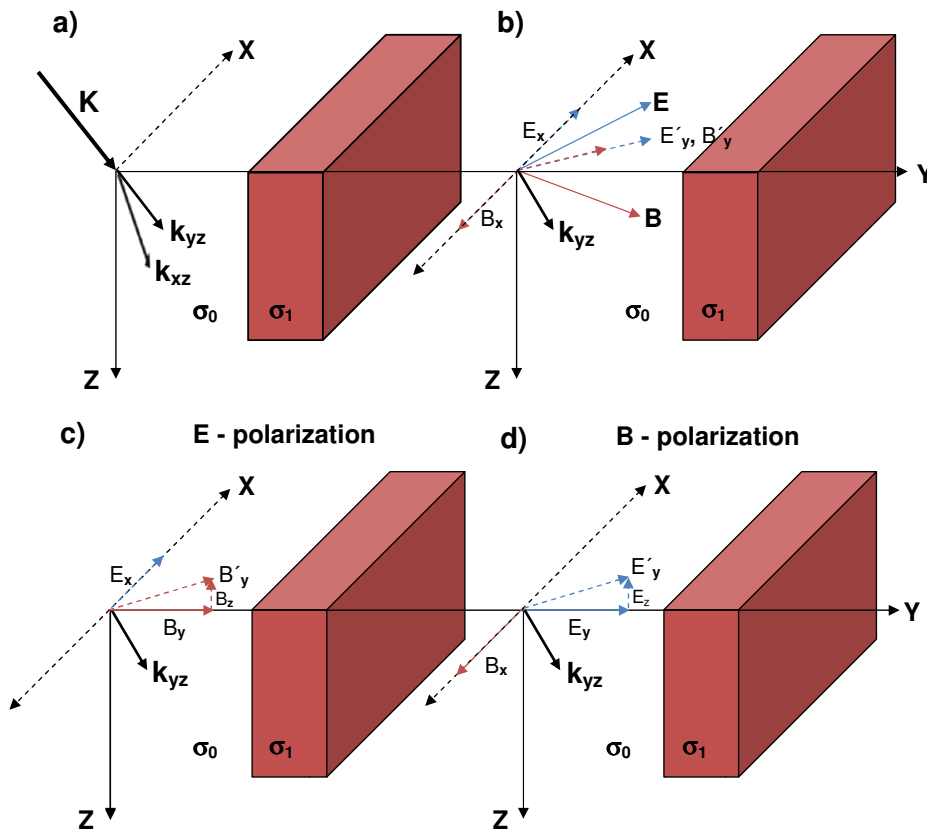


Figure 2.2: Decomposition of a plane electromagnetic wave in 2D earth model. a) incident wavenumber vector \mathbf{K} and decomposition in k_{yz} and k_{xz} in yz- and xz-plane, respectively. b) The associated \mathbf{E} and \mathbf{B} fields and their components. Decomposition of a polarised wave in two waves of c) E-polarisation and d) B-polarisation.

As a first step, we decompose the vector \mathbf{K} in two parts in the xz and yz planes (\mathbf{k}_{xz} and \mathbf{k}_{yz}) (Porstendorfer, 1975). Let us consider the wave associated with the wavenumber vector \mathbf{k}_{yz} . To this vector belongs the magnetic field \mathbf{B} and the electric field \mathbf{E} . Furthermore, \mathbf{E} and \mathbf{B} can be decomposed into their components to the

electric components E_x, E'_y and the magnetic components $B_x, B'_y, 0$, where the prime denotes that the y components of the electric and magnetic fields do not coincide with the y-axis (Fig. 2.2b). The decomposition of the electromagnetic wave in a 2D Earth results in a decoupling of Maxwell's equations into two distinct modes. In the first mode, the current system which is associated with electrical field components E_x is parallel to the strike direction. This mode is usually called transverse electric (TE) or E-polarisation. The electric and magnetic field components associated with this mode are then E_x, B_y and B_z (Fig. 2.2c). In the second mode, the electrical current associated with electrical field component E_y flows perpendicular to the strike direction. This mode is called transverse magnetic (TM) or B-polarisation and the electromagnetic components coupled with this mode are B_x, E_y and E_z (Fig. 2.2c). Considering now the differential form of Maxwell's equation (2.10, 2.11), remembering that $\nabla_x \sigma = 0$ and taking into account only the electromagnetic components that are associated with each mode, Maxwell's equations can be rewritten into two different sets of equations:

E-polarization	B-polarization
$\mathbf{E}(E_x, 0, 0), \mathbf{B}(0, B_y, B_z)$	$\mathbf{E}(0, E_y, E_z), \mathbf{B}(B_x, 0, 0)$
$\frac{\partial B_z}{\partial y} + \frac{\partial B_y}{\partial z} = \mu_0 \sigma E_x$ (2.36a)	$\frac{\partial E_z}{\partial y} + \frac{\partial E_y}{\partial z} = -i\omega B_x$
	$-[(\nabla_{y,z}\sigma) \times \nabla_{y,z} \times \mathbf{B}/\sigma]$ (2.37a)
$-\frac{\partial E_x}{\partial y} = -i\omega B_z$ (2.36b)	$-\frac{\partial B_x}{\partial y} = \mu_0 \sigma E_z$ (2.37b)
$\frac{\partial E_x}{\partial z} = -i\omega B_y$ (2.36c)	$\frac{\partial B_x}{\partial z} = \mu_0 \sigma E_y$ (2.37c)

Taking the first derivatives of eqs. 2.36b and 2.36c with respect to y and z, respectively and substituting in eq. 2.36a, results in the differential equation for E_x :

$$\frac{\partial^2 E_x}{\partial y^2} + \frac{\partial^2 E_x}{\partial z^2} = i\omega \mu_0 \sigma E_x \quad (2.38)$$

Similarly, taking the first derivatives of eqs. 2.37b and 2.37c with respect to y and z, and substituting in eq. 2.37a, gives the differential equation for B_x :

$$\nabla \cdot \rho \nabla B_x = \frac{\partial}{\partial y} \left(\rho \frac{\partial B_x}{\partial y} \right) + \frac{\partial}{\partial z} \left(\rho \frac{\partial B_x}{\partial z} \right) = -i\omega \mu_0 \sigma B_x \quad (2.39)$$

Because of the decoupling of Maxwell's equations in 2D, the impedance tensor presented in eq. 2.30 is reduced to:

$$\begin{pmatrix} 0 & Z_{xy} \\ Z_{yx} & 0 \end{pmatrix} \quad (2.40)$$

Since B_x and B_y vanish for the E- and B-polarisations, respectively, the horizontal impedance tensor transfer functions Z_{xx} and Z_{yy} also vanish.

2.3 Forward modeling of MT data

MT Forward modelling aims to solve Maxwell's equations to simulate the spatial and temporal distribution of electric and magnetic fields in the subsurface for a given conductivity distribution and a range of periods/frequencies.

Several methods and approaches have been suggested over the last four decades to solve Maxwell's equations. For the one dimensional case (1D), there exist analytic solutions to calculate the impedances at the boundaries between layers in a 1D layered earth model and consequently on the surface (Wait recursion formula, (Wait, 1954)). For certain two and three dimensional (2D and 3D) earth models, analytical solutions have been found (Weaver, 1994; Porstendorfer, 1975). These solutions are usually used to verify the accuracy of the numerical solutions.

Several numerical methods exist to solve the induction equations in 2D and 3D. The integral equation method were developed in the 1970ies (i.e. Weidelt, 1975; Wanamaker et al, 1984, Zhdanov et al, 1997). The main concept of the integral equation approach is to solve Maxwell's equations only in sub regions of the model where anomalous (2D or 3D) conductivities exist. To achieve that, the anomalous conductivity structures are divided into rectangular cells (2D) or cubes (3D) while the surrounding structures are considered to be 1D. Considering only regions with anomalous conductivity is possible by a surface (2D) or volume (3D) integration of Maxwell's equations in conjunction with Green's theorems (Zhdanov et al, 1997). Consequently, the total solution of the electromagnetic field is the sum of the normal field which relates to the background conductivity structures and the anomalous fields originating from the anomalous part. Since the numerical solution can be restricted to the anomalous domain, integral equation techniques are computational efficient (i.e. time and storage capacity). The disadvantage of this approach is that a background structure around the anomalous structure has to be fixed (usually a 1D structure).

The most popular and widely used methods to solve Maxwell's equations are the differential equation methods. In these methods the entire earth model (not only the anomalous structures) is subdivided (discretized) into rectangular cells (2D) or cubes (3D) and each cell/cube is assigned a constant conductivity value. For each of these cells/cubes, Maxwell's equations are solved. The main advantage of differential

equation methods is their flexibility in constructing the earth model. A disadvantage is the huge computational requirements caused by fine discretization of the model and consequently having to solve an equation system with a huge number of linear equations to get accurate results.

The solution of the second order partially differential equations (eqs. 2.22 and 2.23) is done by applying either Finite differences (FD) or Finite element (FE) methods. The latter is more flexible in terms of model design since the cells/cubes should not have a rectangular shape which is a requirement for FD method. The FE method is particularly useful if the earth model includes topography or bathymetry.

2.4 2D forward modelling with finite differences

Let us first define a domain Ω of two dimensional anomalous structures with the boundaries $\Gamma_1, \Gamma_2, \Gamma_3$ and Γ_4 coinciding with the top, right, bottom and left sides of Ω , respectively (Fig. 2.3). Let us consider also a Cartesian coordinate system (X,Y and Z) in which Z, Y and X refer to down, east and north, respectively. The solution of Maxwell's equations for the domain Ω can be calculated by approximating the partial differential equations on a staggered grid. In this case, the domain Ω is discretized as a network of cells connecting each other. Each of these cells is assigned a constant conductivity. In the 2D case, the conductivity distribution is defined in the ZY plane, while the XZ plane coincides with the geoelectric strike direction in which the conductivities remain constant ($\frac{\partial\sigma}{\partial x} = 0, \frac{\partial\sigma}{\partial y} \neq 0, \frac{\partial\sigma}{\partial z} \neq 0$). The model is discretized as $i = 0, 1, \dots, M$ lines in Y direction and $j = 0, 1, \dots, N$ lines in Z direction. Thus, the lines 0 and M in the horizontal direction correspond to the left and right boundaries (Γ_2, Γ_4) and lines 0 and N in the vertical direction correspond to the top and bottom boundaries (Γ_1, Γ_3), respectively. The conductivity of each cell in the domain Ω is denoted as σ_{ij} .

Recalling the physical formulation of the induction problem in the previous section, our problem is to find an accurate approximation for

$$\nabla \cdot \nabla E_x = \nabla^2 E_x = \frac{\partial^2 E_x}{\partial y^2} + \frac{\partial^2 E_x}{\partial z^2} = -i\omega\sigma\mu_0 E_x \quad (2.41)$$

for the E-polarization mode, and for

$$\nabla \cdot \rho \nabla B_x = \frac{\partial}{\partial y} \left(\rho \frac{\partial B_x}{\partial y} \right) + \frac{\partial}{\partial z} \left(\rho \frac{\partial B_x}{\partial z} \right) = -i\omega\mu_0 \sigma B_x \quad (2.42)$$

for the B-polarization mode with appropriate boundary conditions. The boundary conditions for the 2D case are discussed in detail by Jones and Price (1969), Weaver and Brewitt-Taylor (1977), and Weaver (1994). Weaver (1994) gives a comprehensive description of the boundary conditions and describes their FD form. Here, just a

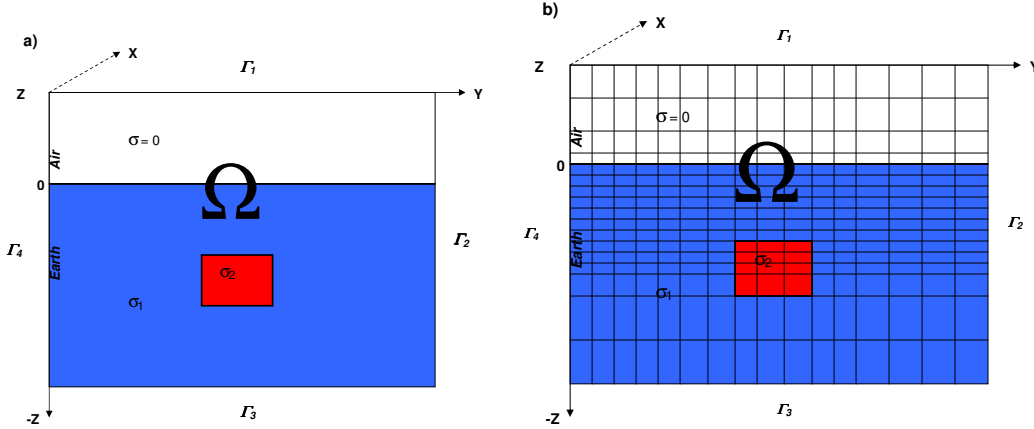


Figure 2.3: a) A simple two dimensional domain Ω consists of the air layer located in the uppermost part of Ω , the lower part represents the solid earth. b) The discretized form of the domain Ω

brief explanation is given. The boundary conditions at the interface between two conductive media (Fig. 2.4) and also at the surface $z = 0$, are:

1. All components (normal and tangential) of the magnetic field (\mathbf{H}) are continuous,

$$\hat{\mathbf{n}} \cdot (\mu_2 \mathbf{H}_2 - \mu_1 \mathbf{H}_1) = 0, \quad \hat{\mathbf{n}} \times (\mu_2 \mathbf{H}_2 - \mu_1 \mathbf{H}_1) = 0$$
2. The tangential components of the electric field (\mathbf{E}) are continuous,

$$\hat{\mathbf{n}} \times (\mathbf{E}_2 - \mathbf{E}_1) = 0$$
3. The normal component of current density (\mathbf{j}) must be continuous across conductivity discontinuities and zero across $z = 0$,

$$\hat{\mathbf{n}} \cdot (\mathbf{j}_2 - \mathbf{j}_1) = \hat{\mathbf{n}} \cdot (\sigma_2 \mathbf{E}_2 - \sigma_1 \mathbf{E}_1) = 0$$
4. The first derivatives of the normal component of \mathbf{E} are continuous.

The last condition implies that $E_z = 0$ inside the conductive earth at $z = 0$. There are also conditions to be satisfied at the boundaries $\Gamma_1, \Gamma_2, \Gamma_3$ and Γ_4 . The most basic condition which can be applied at these boundaries is that the boundaries must be placed far enough from lateral discontinuities so that the fields can be considered uniform in the horizontal directions (Jones and Price, 1970; Jones, 1973).

The solutions of eqs. 2.41 and (2.42) for E_x and B_x , respectively, must be found at each node in the sub-region $\Omega_i \subset \Omega$. At the same time the internal and external boundary conditions must be satisfied. The explanation below will cover the solution of eq. 2.41 for E_x . The same concept can be applied to solve eq. 2.42 for B_x .

Let us consider a typical nodal point i, j located in the inner part of a rectangular, but not necessarily uniform, grid (Fig. 2.5). Its neighboring nodes are $i, j+1$, $i, j-1$, $i-1, j$ and $i+1, j$, which are located to the right, left, down and up from i, j , respectively.

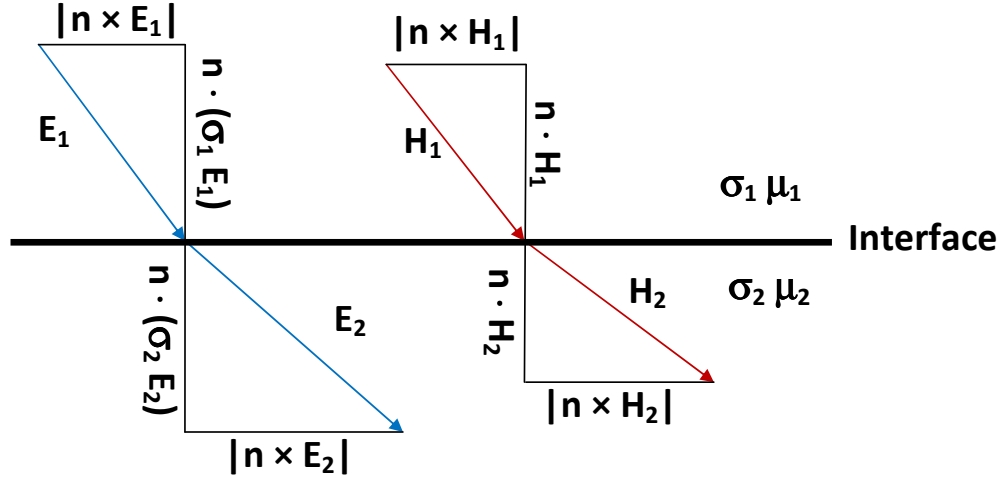


Figure 2.4: Boundary conditions of electromagnetic field, at an interface between two regions.

By integrating eq. 2.41 over the rectangular area A around i,j we find E_x at i,j . By making use of Gauss's theorem, we can convert the integral of the divergence ($\nabla \cdot$) over an area to an integral of the gradient (∇) along a line. Thus, the left hand side of eq. 2.41 is rewritten as:

$$\int_A \nabla \cdot \nabla E_x dA = \int_L \hat{\mathbf{n}} \cdot \nabla E_x dL \quad (2.43)$$

where, $\hat{\mathbf{n}}$ is the outward unit normal vector to the edges of the area A (green rectangular in Fig. 2.5) and L is the line surrounding A (Roman numbers in Fig. 2.5).

From the boundary condition number 4 follows that the first derivatives of E_x are continuous everywhere. Consequently, we can accurately approximate derivatives normal to the line of integration in eq. 2.43 using centered first differences. For instance, let us consider the right edge of A . This edge consists of two pieces (II and III in Fig. 2.5) of the line L . Thus, the gradient of the outward normal from this edge to the right is approximated by:

$$\int_l \frac{\partial E_x}{\partial y} dl \approx \left(\frac{\Delta z_{i-1} + \Delta z_i}{2} \right) \frac{E_x^R - E_x^P}{\Delta y_j} \quad (2.44)$$

where, E_x^R and E_x^P are the E_x values at the right node and at the node under consideration (i,j), respectively. Adding the contribution from the four sides of A , the line

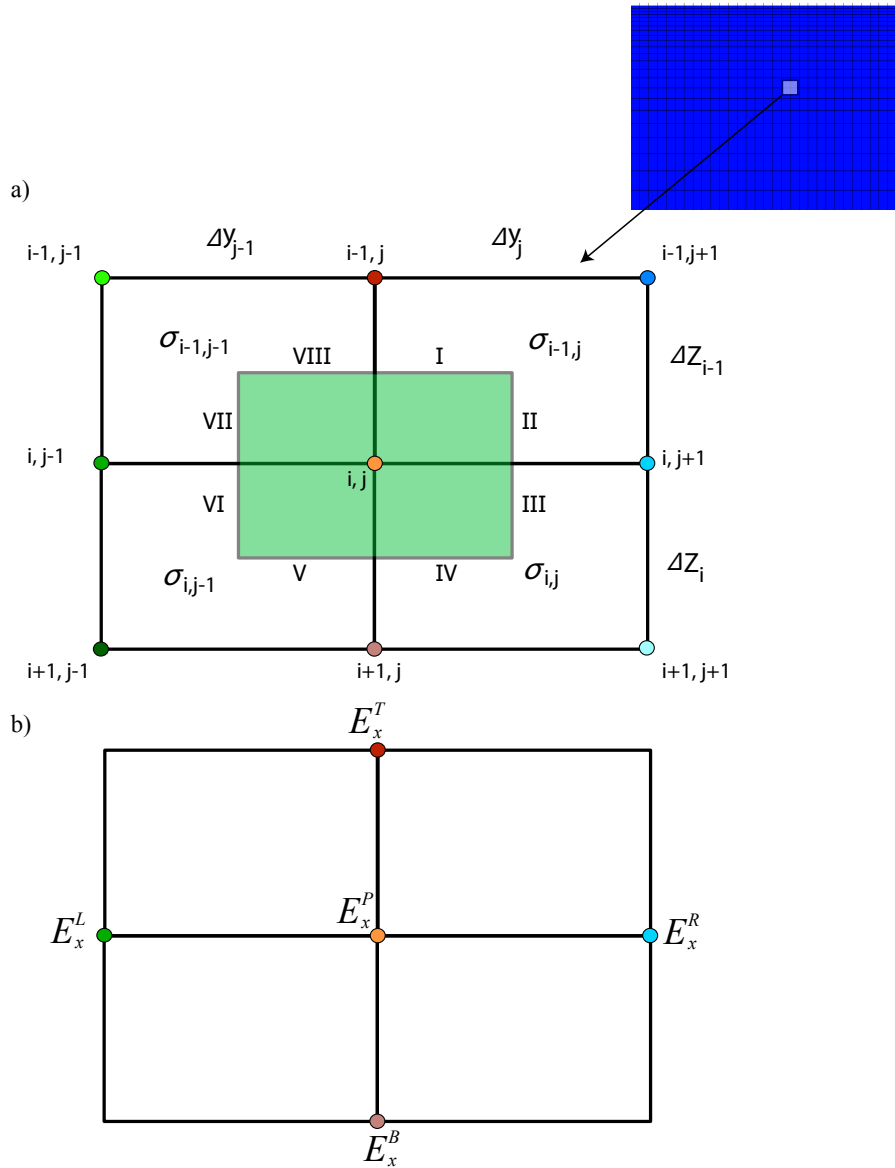


Figure 2.5: a) A five points stencil scheme used in approximating b) the electric field component E_x^P using the right (E_x^R), left (E_x^L), upper (E_x^T) and bottom (E_x^B) E_x values.

integral in eq. 2.43 becomes:

$$\begin{aligned}
\int_l \hat{n} \cdot \nabla E_x dL &\approx \left(\frac{\Delta z_{i-1} + \Delta z_i}{2} \right) \frac{E_x^R - E_x^P}{\Delta y_j} \\
&+ \left(\frac{\Delta z_{i-1} + \Delta z_i}{2} \right) \frac{E_x^L - E_x^P}{\Delta y_{j-1}} \\
&+ \left(\frac{\Delta y_{j-1} + \Delta y_j}{2} \right) \frac{E_x^U - E_x^P}{\Delta z_i} \\
&+ \left(\frac{\Delta y_{j-1} + \Delta y_j}{2} \right) \frac{E_x^D - E_x^P}{\Delta z_{i-1}}
\end{aligned} \tag{2.45}$$

The left hand side of eq. 2.41 is also integrated over the area A, with the result:

$$-\int_A i\omega\mu_0 E_x(y, z)\sigma(y, z)dA = -i\omega\mu_0 E_x^P \int_A \sigma(y, z)dA \tag{2.46}$$

The surface integral over $\sigma(y, z)$ in eq. 2.46 is calculated by summing up the product of conductivity at each cell around the node (i,j) with the cell dimensions. Thus, the left hand side of eq. 2.46 takes the form:

$$\begin{aligned}
-i\omega\mu_0 E_x^P \int_A \sigma(y, z)dA &\approx -i\omega\mu_0 E_x^P [\sigma_{i,j}\Delta z_i\Delta y_j + \sigma_{i-1,j}\Delta z_{i-1}\Delta y_j \\
&+ \sigma_{i,j-1}\Delta z_i\Delta y_{j-1} + \sigma_{i-1,j-1}\Delta z_{i-1}\Delta y_{j-1}] \\
&= -i\omega\mu_0 R(\sigma, \Delta z, \Delta y) E_x^P
\end{aligned} \tag{2.47}$$

where $R(\sigma, \Delta z, \Delta y)$ is a function that depends on σ and the cell dimension. Equating eqs. 2.45 with 2.47 yields after rearrangements:

$$C_L^{ij} E_x^L + C_R^{ij} E_x^R, C_T^{ij} E_x^T + C_B^{ij} E_x^B = C_P^{ij} E_x^P \tag{2.48}$$

where

$$\begin{aligned}
C_R^{ij} &= \left(\frac{\Delta z_{i-1} + \Delta z_i}{2\Delta y_j} \right) \\
C_L^{ij} &= \left(\frac{\Delta z_{i-1} + \Delta z_i}{2\Delta y_{j-1}} \right) \\
C_T^{ij} &= \left(\frac{\Delta y_{j-1} + \Delta y_j}{2\Delta z_{i-1}} \right) \\
C_B^{ij} &= \left(\frac{\Delta y_{j-1} + \Delta y_j}{2\Delta z_i} \right) \\
C_P^{ij} &= -i\omega\mu_0 R(\sigma, \Delta z, \Delta y).
\end{aligned}$$

Here, C_R^{ij} , C_L^{ij} , C_T^{ij} and C_B^{ij} are the coupling coefficients to the node (i,j) from right, left, top and bottom nodes, respectively. C_P^{ij} is the self coupling coefficient. This

formulation of the FD equations does not include the error term ($O(\Delta)$) which originates from the numerical formulation of the problem. This term depends mainly on the gridding strategy used to discretize the region Ω . The effect of the discretization on the approximated solution is discussed and analysed in [Smith \(1996a\)](#).

Separating the known terms (cells dimensions and conductivities) from the desired unknown terms (E_x on all nodes) in eq. 2.48, the finite differences equations and the boundary conditions can be formulated in matrix-vector notation as:

$$\mathbf{Ax} = \mathbf{b} \quad (2.49)$$

where \mathbf{A} is a complex non-Hermitian matrix containing the coupling coefficients, \mathbf{x} is the desired solution vector and \mathbf{b} is a vector describing the boundary values.

Solving eq. 2.49 for \mathbf{x} in the TE-mode yields the total electric field \mathbf{E}_x on all nodes located in Ω . While in the TM-mode, the solution \mathbf{x} corresponds to the total magnetic field \mathbf{B}_x . The auxiliary field components $\mathbf{B}_y, \mathbf{B}_z$ in the TE-mode and $\mathbf{E}_y, \mathbf{E}_z$ in the TM-mode are derived from the total field components in each mode by taking their first derivatives in horizontal and vertical directions using eqs. 2.36b, 2.36c, 2.37b and 2.37c ([Weaver, 1994](#)).

The equation system (2.49) is fundamental for solving the forward modelling and inversion problems of MT, because the solution vector \mathbf{x} is used in computing the model responses and the sensitivity matrix for the inversion. Thus, considerable effort must be taken into account when solving eq. 2.49. The equation system presented in eq. 2.49 has many advantageous properties that can be used to accelerate the numerical solution:

1. The matrix \mathbf{A} is a complex non-Hermitian, sparse and banded matrix. The only complex elements of the matrix are located on the main diagonal. In each row of \mathbf{A} there are 5 non-zero elements in the 2D case. To better understand the structure of the matrix \mathbf{A} , let us consider a simple example. The 2D model in which we desire to approximate the Maxwell's equations consists of nine model parameters numbered from 1 to 9 ($\sigma_1, \dots, \sigma_9$). In total, there are 16 nodes for which the solution of eq. 2.49 for \mathbf{x} must be computed ([Fig. 2.6](#)).

In this example the dimension of the matrix \mathbf{A} is 16×16 . The main diagonal of \mathbf{A} contains the central point coefficients $C_P^i, i = 1, \dots, 16$. The off diagonal elements in each row correspond to the neighbouring nodes of the central node. The location of each non-zero element in the matrix must coincide with its corresponding node in the model. For example, the right neighbour of the node number 1 is node number 5, thus, the 5th column in the first row contains the value of C_R^1 , etc.

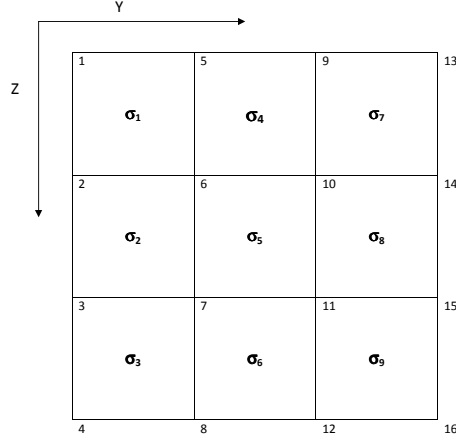


Figure 2.6: The example 2D model contains 9 model parameters ($\sigma_1, \dots, \sigma_9$) resulting in 16 nodes for which the solution of equation 2.49 must to be computed. This example model is used to explain the structure of the coupling coefficients matrix \mathbf{A} .

$$\begin{pmatrix} C_P^1 & C_B^1 & 0 & 0 & C_R^1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ C_T^2 & C_P^2 & C_B^2 & 0 & 0 & C_R^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & C_T^3 & C_P^3 & C_B^3 & 0 & 0 & C_R^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & C_T^4 & C_P^4 & 0 & 0 & 0 & C_R^4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & C_P^5 & C_B^5 & 0 & 0 & C_R^5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & C_L^6 & 0 & 0 & C_T^6 & C_P^6 & C_B^6 & 0 & 0 & C_R^6 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_L^{16} & 0 & 0 & C_T^{16} & C_P^{16} \end{pmatrix}$$

2. Since Maxwell's equations must be approximated for a set of periods, eq. 2.49 must be solved for each period. It is worthwhile to note that only the complex main diagonal elements (in particular the imaginary part) depends on a particular period. This means that the off-diagonal elements of \mathbf{A} must be computed only once and the main diagonal can be computed for every period separately.
3. A very important point in solving eq. 2.49 for a set of periods is to note that the solution vector \mathbf{x} for one particular period is independent from the solution vector \mathbf{x} of another period. However, in case of using an iterative solver to solve eq. 2.49, it is possible to use the solution vector \mathbf{x} for one period as a starting value for the next period. This is due to the fact that the EM field changes smoothly from one period to another (Egbert, 2006). Furthermore, the solutions of eq. 2.49 for E_x and B_x in TE- and TM-mode, respectively, are independent from each other.

2.5 Forward modelling in 3D

The solution of the electromagnetic induction problem in three dimensions is not a straightforward task when compared to the solution of the problem in two dimensions. This is because the conductivity of the earth varies in three spatial directions ($\sigma(x, y, z)$). Furthermore, the 3D induction theory allows the source field to have an arbitrary geometry. The source field geometry has a remarkable effect when the earth model is very large, so that the assumption that the EM field is horizontal and uniform is not valid anymore (Weaver, 1994). However, for simplicity, to solve the 3D induction problem we will assume that only the conductivity of the earth varies in three directions and keep the source field horizontal and uniform.

The solution of Maxwell's equations in 3D is best done numerically by either solving the second order equation:

$$\nabla \times \nabla \times \mathbf{E} = i\omega\mu_0\sigma\mathbf{E} - \nabla[(\nabla\sigma) \cdot \mathbf{E}/\sigma] \quad (2.50)$$

for \mathbf{E} or solving:

$$\nabla \times \nabla \times \mathbf{B} = i\omega\mu_0\sigma\mathbf{B} - [(\nabla\sigma) \times (\nabla \times \mathbf{B})/\sigma] \quad (2.51)$$

for \mathbf{B} . In comparison to the solution of the induction problem in 2D, the solution of eq. 2.50 or 2.51 is obtained in terms of a three components vector field (E_x, E_y, E_z or B_x, B_y, B_z) rather than one scalar component. Furthermore, eq. 2.50 for the electric field (\mathbf{E}) contains a term involving the conductivity gradient which does not appear in 2D. Nevertheless, applying the finite differences (FD) technique to approximate eq. 2.50 or 2.51 in a 3D grid follows roughly the same concept as for 2D case. This means that the earth model is first discretized in cubes (rather than cells as in 2D) and each cube is assigned to a constant conductivity. Afterwards, an approximation of the EM field is computed at each edge and face of the cubes rather than at the nodes of the cells as in 2D.

In general there are two ways to define the numerical grid for the FD approximation. In the first type of the 3D numerical grid, the three electric field components are defined at the edges of the cubes while the three magnetic field components are defined at the faces of cubes (Fig 2.7a). In this case a solution of eq. 2.50 for \mathbf{E} is sought and the secondary magnetic field is then derived from \mathbf{E} using the first Maxwell equation (eq. 2.1, Farady's law). This specification of the EM field on staggered grids were used by Yee (1966) for solving the boundary value problems of Maxwell's equations. Because of that, this kind of numerical grid is sometimes called Yee-Grid. This type of the numerical grid was used later by Alumbaugh *et al.* (1996), Newman & Alumbaugh (1997) for wideband electromagnetic 3D modelling and by Siripunvaraporn *et al.* (2005); Egbert (2006) for 3D MT. In the second type of the numerical grid, the magnetic field components are defined at the edges of the cubes

and the secondary electric field components are defined at the faces of the cubes (Fig 2.7b). In this case, eq. 2.51 is solved for \mathbf{B} and \mathbf{E} is derived from the second Maxwell equation (eq. 2.2, Ampere's law). This type of 3D grid has been used by Mackie *et al.* (1994) for 3D MT modelling and by Smith (1996b). A comparison between these two types of numerical grids, concerning their numerical accuracy and speed in solving the forward modeling problem in 3D MT, was discussed by Siripunvaraporn *et al.* (2002).

Independently from the used numerical grid to solve either eq. 2.50 or 2.51 using

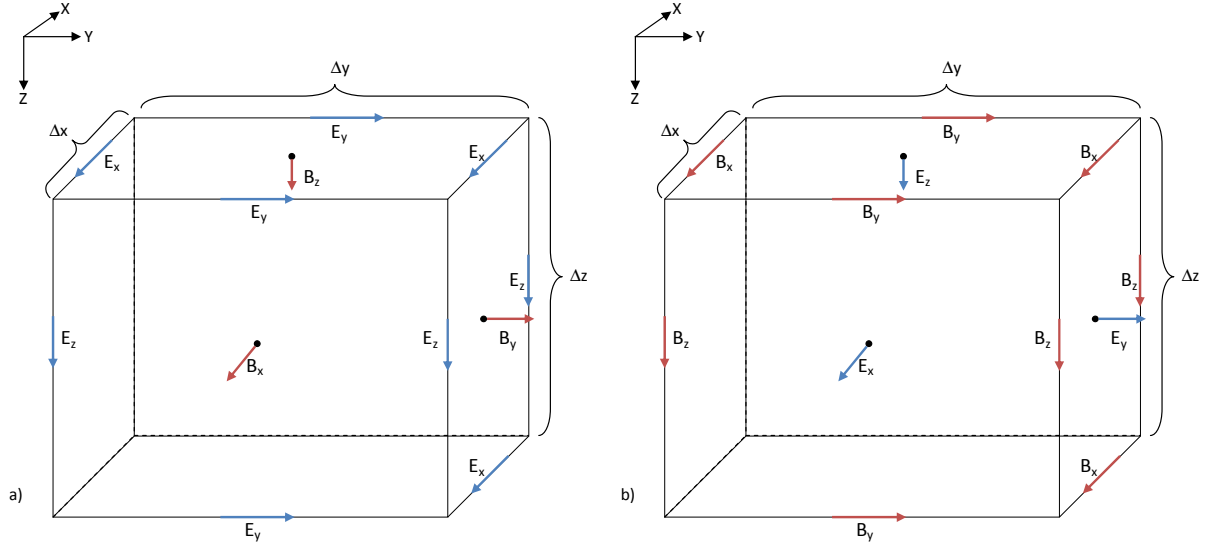


Figure 2.7: Two different definitions of the numerical grids used in the finite differences method to sample a) the primary electric field components at the edges of the cube and the secondary magnetic field components at the faces. b) The convention used to sample the primary magnetic field components at the edges of the cube and the secondary electric field components at the faces. Δx , Δy and Δz are the dimension of the cube in x,y and z directions, respectively.

FD, we usually obtain a linear equation system:

$$\mathbf{Ax} = \mathbf{b} \quad (2.52)$$

which is similar to the equations system obtained in 2D. However, the dimension of \mathbf{A} , \mathbf{x} and \mathbf{b} are different. The complex, non-Hermitian and sparse matrix \mathbf{A} contains the value of the coupling coefficients between a central edge and its 12 neighbouring edges. For example, the 12 thick-lined edges in fig. 2.8 are required to form the corresponding equation for E_x at node (i,j,k). This means, the corresponding row in the coefficients matrix \mathbf{A} contains 13 non-zero elements (12 neighbouring edges plus the central one). A similar procedure is used for other edges in the grid and other electric field components. This results in a coefficients matrix, and hence an equations system,

which has the dimension of $N_x(N_y+1)(N_z+1)+(N_x+1)N_y(N_z+1)+(N_x+1)(N_y+1)N_z$, where N_x , N_y and N_z are the numbers of model parameters in x,y and z directions, respectively. More details on how to represent the operators $\nabla \times \nabla \times \mathbf{E}$ or $\nabla \times \nabla \times \mathbf{B}$ in a discrete form to compute the corresponding coupling coefficients in 3D and therefore constructing the matrix \mathbf{A} are given i.e. in Weaver (1994), Alumbaugh *et al.* (1996), Smith (1996a) and Mackie (1996).

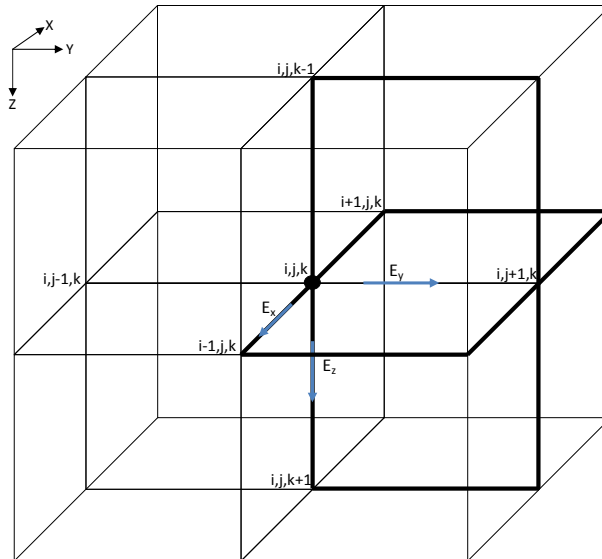


Figure 2.8: Small part of the staggered grid used to compute the electric field components E_x , E_y and E_z at the edges of the cubes. This figure is used to emphasise the dependency of the edges on each other to construct the coefficients matrix \mathbf{A} . For example, to form the equations needed to compute the electric field component E_y which associated with the node (i,j,k) , the 12 electric field components locate at the edges marked with thick lines are needed.

Solving eq. 2.52 for \mathbf{x} results in the values of the three electric field components E_x , E_y and E_z at all edges of the grid. Using Faraday's law (eq. 2.84) the magnetic field components are derived numerically from the electric field components.

The three components of the electric field are calculated for a current system which flows in N-S (first mode) or in E-W (second mode) direction. To obtain the same electric field components, however, for another orientation of the current system, we can rotate the model by 90° , to construct the equations system (2.52) and solving it again for \mathbf{x} (Weaver, 1994). The need for the three electric field components using from two orientations of the current system is based on the fact that the model responses in 3D MT are the full impedance tensor and the magnetic vertical transfer functions. Thus, to construct the full impedance tensor we need two orientations of

the current system, so that:

$$\begin{pmatrix} Z_{xx} & Z_{xy} \\ Z_{yx} & Z_{yy} \end{pmatrix} = \begin{pmatrix} E_x^1 & E_x^2 \\ E_y^1 & E_y^2 \end{pmatrix} \begin{pmatrix} B_x^1 & B_x^2 \\ B_y^1 & B_y^2 \end{pmatrix}^{-1} \quad (2.53)$$

where the subscripts 1 and 2 denote the electric or magnetic component arising from first and second modes, respectively.

The right hand side vector \mathbf{b} in eq. 2.52 contains the values of the boundary conditions. For 3D MT forward modelling the boundary values are those which correspond to the northernmost, southernmost, easternmost, westernmost, surface and bottom faces of the staggered grid. To satisfy the boundary conditions at these faces of the 3D grid, the conductivity reaches its 2D distribution ($\sigma(y, z)$ or $\sigma(x, z)$) at these faces and its 1D distribution ($\sigma(z)$) at the edges of these faces. In another words, to solve the 3D MT forward modelling problem first we have to solve the problem in 1D and then in 2D and assign the computed values to the corners or faces of the 3D grid, respectively. For solving the 2D problem at the end faces of the 3D grid we have to distinguish between solving the TE- or TM-modes problems depending on the current system which flows in the 3D grid. For example, if the current system flows in N-S direction then we have to solve the 2D TE-mode problem at the northernmost and southernmost faces and TM-mode problem at the easternmost and westernmost faces.

Apart from the complications that may arise from constructing the matrix \mathbf{A} and the vector \mathbf{b} to form the equation system presented in eq. 2.52, we are interested to observe the properties of eq. 2.52 to use them later for solving the 3D MT forward modelling problem on massively parallel computers (section 3.3).

The equation system in 3D (eq. 2.52) has similar properties as the equation system in 2D (eq. 2.49), however, the dimensions differ in both equations. Furthermore, solving eq. 2.49 in 2D for TE- or TM-modes is completely independent of each other. This means that model responses in 2D are computed from electric and magnetic components result from one mode without having to solve the problem in the other mode. In the 3D case the situation is different: In order to compute the model responses the electric and magnetic fields components result from two orientations of the current system are required (eq. 2.53). However, it is important to note that this dependency of the modes in 3D forward modelling is only required when computing the model responses and not when computing the electric and magnetic components for each mode. This means that the computation process to obtain the required electric and magnetic components from each mode can be done in parallel. Furthermore, as in the 2D case, the solution of the eq. 2.52 is obtained for a set of periods, which can also can be done in parallel, since the solution vector \mathbf{x} for one period dose not depend on the solution vector from another period, unless we want to use the previous solution vector \mathbf{x} as a starting value to solve for the next period.

2.6 Inversion of magnetotelluric data

The ultimate goal of inversion theory is to provide a mathematical framework to transform measured data from the data space to the model space in order to estimate model parameters. Solving such inverse problems arises in many branches of the medical, physical and geophysical sciences. In geophysics, solving the inverse problem aims to determine the structure of the earth. Depending on the used geophysical method, the structure of the earth can be explained using the distribution of the electrical resistivity (geoelectric and electromagnetic methods), acoustic velocity (seismic and seismology), density (gravimetry), etc. These physical properties of the rocks are a reflection of the geological formations, which define the structure of the earth.

The most common and widespread inversion approaches can be classified in two main categories:

- Linearized inversion: A non linear function mapping the model parameters from model space to data space (the forward modelling operator) is first linearized before the inverse problem is solved.
- Non linear inversion: The inverse problem is solved directly using the non linear forward modelling operator.

Each of the above inversion schemes can be solved in model or in data space. The difference between both methods, as we will see later, is in the size of the inverse problem, which becomes considerably important when inverting geophysical data in two and three dimensions.

2.6.1 Linearized inversion in the model space

In the inversion process of electromagnetic data, particularly MT, we have some measured data in form of horizontal and/or vertical transfer functions or apparent resistivities and phases for which we try to find model parameters (resistivities) that can explain the measured data. Before describing the mathematical formulations for the inversion process, let us set a few notations that will be used throughout the text:

- We can treat N measured data as elements of a data vector \mathbf{d} ,

$$\mathbf{d} = [d_1, d_2, d_3, \dots, d_N]^T$$

The N -dimensional data vector resides in the data space. Usually, all measured data are contaminated with an error vector \mathbf{e} .

- The model \mathbf{m} we are seeking contains M model parameters and can be presented as:

$$\mathbf{m} = [m_1, m_2, m_3, \dots, m_M]^T$$

The model vector resides in the model space.

- The projection from model space to data space is achieved by the forward modelling operator \mathbf{F} , which is function of \mathbf{m} . The operator $\mathbf{F}(\mathbf{m})$ generates the predicted or synthetic data. The model parameter vector \mathbf{m} and the data vector \mathbf{d} are related to each other via:

$$\mathbf{F}(\mathbf{m}) = \mathbf{d} + \mathbf{e}$$

The inversion process attempts to find a model that reproduces the measured data within their error. In other words, the inversion process tries to minimise the misfit or residuum between measured and predicted data. To assess the quality of the fit, a weighted least squares criterion is used (Jackson, 1972; Constable *et al.*, 1987),

$$\chi^2 = \sum_{i=1}^N \frac{(d_i - F_i(m))^2}{e_i^2} \quad (2.54)$$

where e_i is the error of the i 'th data point. In a least squares sense, the function to be minimised is known as misfit-function, cost-function, objective function, penalty function, or energy function, is defined as,

$$\Psi_d(\mathbf{m}) = \|\mathbf{W}\mathbf{d} - \mathbf{W}\mathbf{F}(\mathbf{m})\|^2 = [\mathbf{d} - \mathbf{F}(\mathbf{m})]^T \mathbf{W}^T \mathbf{W} [\mathbf{d} - \mathbf{F}(\mathbf{m})] \quad (2.55)$$

where \mathbf{W} is the diagonal $N \times N$ matrix which contains the inverse of the data errors,

$$\mathbf{W} = \text{diag}\{1/e_1, 1/e_2, \dots, 1/e_N\}$$

Since \mathbf{W} is a diagonal matrix, the term $\mathbf{W}^T \mathbf{W}$ can be eliminated from the definition of $\Psi_d(\mathbf{m})$ by rescaling the measured and predicted data (Egbert, 2006).

From mathematical point of view, the function $\Psi_d(\mathbf{m})$ can be minimised using a least square approach. However, this is defined for linear functions. The operator \mathbf{F} is a function of \mathbf{m} , however, this functionality is not linear. Hence, a non-linear least square approach must be applied. This means, that the function $\mathbf{F}(\mathbf{m})$ must first be linearized. The linearization of the forward modelling operator can be achieved by a first order Taylor-expansion around a starting, or a so-called initial guess model (\mathbf{m}_0). In this case, the linearized $\mathbf{F}(\mathbf{m})$ function can be written as,

$$\mathbf{F}(\mathbf{m}_0 + \Delta\mathbf{m}) = \mathbf{F}(\mathbf{m}_0) + \frac{\partial \mathbf{F}(\mathbf{m}_0)}{\partial \mathbf{m}_0} \Delta\mathbf{m} = \mathbf{F}(\mathbf{m}_0) + \mathbf{J} \Delta\mathbf{m} \quad (2.56)$$

where \mathbf{J} is the sensitivity or Jacobian matrix and $\Delta\mathbf{m}$ a small perturbation around \mathbf{m}_0 . As the name suggests, \mathbf{J} describes how sensitive the predicted data are towards a small changes in the model parameters. As shown in eq. 2.56, the elements of this

matrix correspond to the first derivative of the predicted data with respect to the model parameters,

$$\mathbf{J} = \begin{pmatrix} \frac{\partial F_1(m)}{\partial m_1} & \frac{\partial F_1(m)}{\partial m_2} & \dots & \frac{\partial F_1(m)}{\partial m_M} \\ \frac{\partial F_2(m)}{\partial m_1} & \ddots & \dots & \vdots \\ \vdots & \dots & \ddots & \vdots \\ \frac{\partial F_N(m)}{\partial m_1} & \dots & \dots & \frac{\partial F_N(m)}{\partial m_M} \end{pmatrix} \quad (2.57)$$

Returning to our minimization problem, perturbing \mathbf{m}_0 with $\Delta\mathbf{m}$, the penalty function $\Psi_d(\mathbf{m})$ in eq. 2.55 takes the form,

$$\Psi_d(\mathbf{m}_0 + \Delta\mathbf{m}) = [\mathbf{d} - \mathbf{F}(\mathbf{m}_0 + \Delta\mathbf{m})]^T [\mathbf{d} - \mathbf{F}(\mathbf{m}_0 + \Delta\mathbf{m})] \quad (2.58)$$

Substituting eq. 2.56 in eq. 2.58 yields,

$$\begin{aligned} \Psi_d(\mathbf{m}_0 + \Delta\mathbf{m}) &= [\mathbf{d} - \mathbf{F}(\mathbf{m}_0) - \mathbf{J}\Delta\mathbf{m}]^T [\mathbf{d} - \mathbf{F}(\mathbf{m}_0) - \mathbf{J}\Delta\mathbf{m}] \\ &= [\mathbf{d}' - \mathbf{J}\Delta\mathbf{m}]^T [\mathbf{d}' - \mathbf{J}\Delta\mathbf{m}] \end{aligned} \quad (2.59)$$

where,

$$\mathbf{d}' = \mathbf{d} - \mathbf{F}(\mathbf{m}_0)$$

describes the misfit between measured and predicted data. An extremal stationary point (minimum or maximum) of the penalty function $\Psi_d(\mathbf{m})$ can be found by taking the derivative of eq. 2.59 with respect to $\Delta\mathbf{m}$, and setting the result equal to zero,

$$\begin{aligned} \frac{\partial \Psi_d(\mathbf{m}_0 + \Delta\mathbf{m})}{\partial \Delta\mathbf{m}} &= \frac{\partial}{\partial \Delta\mathbf{m}} ([\mathbf{d}' - \mathbf{J}\Delta\mathbf{m}]^T [\mathbf{d}' - \mathbf{J}\Delta\mathbf{m}]) = 0 \\ &= \frac{\partial}{\partial \Delta\mathbf{m}} (\mathbf{d}'^T \mathbf{d}' - \mathbf{d}'^T \mathbf{J}\Delta\mathbf{m} - \Delta\mathbf{m}^T \mathbf{J}^T \mathbf{d}' + \Delta\mathbf{m}^T \mathbf{J}^T \mathbf{J}\Delta\mathbf{m}) = 0 \\ &= 2\mathbf{J}^T \mathbf{d}' - 2\mathbf{J}^T \mathbf{J}\Delta\mathbf{m} = 0 \\ \mathbf{J}^T \mathbf{J}\Delta\mathbf{m} &= \mathbf{J}^T \mathbf{d}' \\ \Delta\mathbf{m} &= (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \mathbf{d}' \end{aligned} \quad (2.60)$$

Equation 2.60 represent the so-called normal equations, which is also known as the Gauss-Newton or unconstrained least square solution. In this algorithm, eq. 2.60 is solved for the model update $\Delta\mathbf{m}$ and added to the starting model \mathbf{m}_0 . Since the Taylor-expansion to linearize the forward modelling operator in eq. 2.56 is only an approximation¹, an iterative procedure is sought to improve \mathbf{m}_0 . The usual procedure used in the Gauss-Newton algorithm can be addressed as follows:

1. Solving the forward modelling problem and computing χ^2 using eq. 2.54.
2. If χ^2 is greater than a desired value, then compute the sensitivity matrix and solve the normal equations (eq. 2.60).

¹In the Taylor-expansion, the higher order derivatives have been ignored.

3. Steps 1 and 2 are repeated until convergence is reached (i.e. the desired value of χ^2 or the number of iterations is exceed).

The convergence of the Gauss-Newton algorithm to the stationary point of $\Psi_d(\mathbf{m})$ (global minimum) is very slow, if it convergences at all. This is due to the fact that:

- The starting model (\mathbf{m}_0) is far away from the “true” model. To overcome this problem it is necessary that various geological and other geophysical information exist which would allow constructing a proper starting model.
- If $\mathbf{J}^T \mathbf{J}$ is singular ($\det(\mathbf{J}^T \mathbf{J})=0$), the inverse of $\mathbf{J}^T \mathbf{J}$ does not exist, and hence, there will be no solution for eq. 2.60. Furthermore, if $\mathbf{J}^T \mathbf{J}$ is almost singular ($\det(\mathbf{J}^T \mathbf{J}) \ll 1$) then, the solution will not converge.

The problem of having to deal with the singularity of $\mathbf{J}^T \mathbf{J}$ is common for almost all geophysical methods, because we usually need many more model parameters to explain the N -dimensional data vector ($N \ll M$). This means that many columns of \mathbf{J} are zero or almost zero. To circumvent the ill posed problem ($\det(\mathbf{J}^T \mathbf{J}) \ll 1$) in the Gauss-Newton method, constraints are often imposed on the normal equations. Applying constraints on the normal equations implies that the model must have certain characteristics. Possible constraints are:

- The model update vector $\Delta \mathbf{m}$ must have a small norm,

$$\min. \quad \|\Delta \mathbf{m}\|^2 \quad (2.61)$$

This condition also ensures that the Taylor-expansion is always valid for small values of $\Delta \mathbf{m}$.

- The model is constraint so that the difference between two adjacent model parameters will be at its minimum. This can be achieved by using the roughness matrix \mathbf{R}

$$\min. \quad \|\mathbf{R} \mathbf{m}\|^2 \quad (2.62)$$

- The model must consider a priori information in the form of model (\mathbf{m}^{apri}),

$$\min. \quad \|\mathbf{m} - \mathbf{m}^{apri}\|^2 \quad (2.63)$$

This condition ensures that the final model will be close to the a priori model.

Using any of the previous conditions must be considered along with penalty function $\Psi_d(\mathbf{m})$. In this case, we can write a penalty function which considers the original minimization problem (eq. 2.55) and one of the previous constraints. For example,

$$\min. \quad \|\mathbf{d} - \mathbf{F}(\mathbf{m})\|^2 \quad \text{and} \quad \min. \quad \|\Delta \mathbf{m}\|^2$$

or,

$$\Psi = \Psi_d(\mathbf{m}) + \lambda\Psi_m(\mathbf{m}) \quad (2.64)$$

where λ is the so called regularization or trade-off parameter. A suitable value of λ ensures that neither $\Psi_d(\mathbf{m})$ nor $\Psi_m(\mathbf{m})$ will preponderate in the penalty function Ψ : Both will be evenly minimized. Thus, an optimised value of λ must be sought. The relationship between $\Psi_d(\mathbf{m})$ and $\Psi_m(\mathbf{m})$ for various values of λ takes the form of an L-curve if we plot $\Psi_d(\mathbf{m})$ against $\Psi_m(\mathbf{m})$ in a log-log plot. Mathematically, this can be described as follows,

$$\begin{aligned} \lim_{\lambda \rightarrow 0} \Psi_d &\rightarrow \min \quad \text{and} \quad \lim_{\lambda \rightarrow 0} \Psi_m \rightarrow \max \\ \lim_{\lambda \rightarrow \infty} \Psi_d &\rightarrow \max \quad \text{and} \quad \lim_{\lambda \rightarrow \infty} \Psi_m \rightarrow \min \end{aligned}$$

This indicates, that our original minimisation problem turns to minimization and optimization with regard to the regularization parameter.

Marquardt-Levenberg inversion

One possibility to prevent the solution of the normal equations (eq. 2.60) from divergence was introduced by [Levenberg \(1944\)](#). He modified eq. 2.60 to take the form:

$$\Delta\mathbf{m} = (\mathbf{J}^T\mathbf{J} + \beta\mathbf{I})^{-1}\mathbf{J}^T\mathbf{d}' \quad (2.65)$$

where \mathbf{I} is the identity matrix and β is the so-called damping factor. Eq. 2.65 can be derived if we minimize eq. 2.64 following the same procedure used to derive eq. 2.60. The additional term $(\beta\mathbf{I})$ in eq. 2.65 restricts the length of the model update vector to a certain length depending on the choice of β . The solution by the Marquardt-Levenberg method is constructed using mainly eigenvectors of $\mathbf{J}^T\mathbf{J}$, which correspond to high eigenvalues ([Jupp and Vozoff, 1975](#)). This method is also known as *Damped Gauss-Newton* ([Madsen et al., 2004](#)). Similar iterative procedures as in the Gauss-Newton method must be applied to improve the initial guess (\mathbf{m}_0). However, a suitable method to set the initial value and to compute β must also be considered.

Occam Inversion

Another widespread inversion algorithm was introduced by [Constable et al. \(1987\)](#) is the so-called Occam inversion. The idea is to find a model which shows just as much structure as is needed to explain the data. The Occam (or minimum structures) inversion imposes a smoothness constraint on the model. In this case the constraint term, which should be considered in the penalty function, has the form,

$$\|\mathbf{Rm}\|^2 \quad (2.66)$$

The matrix \mathbf{R} represents the so-called roughness matrix. In a uniform grid, the elements of $\mathbf{R}\mathbf{m}$ are defined as the Laplacian of the model parameters which describe the differences between adjacent model parameters (Rodi & Mackie, 2001). In the 1D case, \mathbf{R} takes the form of (Constable *et al.*, 1987),

$$\mathbf{R} = \begin{pmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & 0 & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix} \quad (2.67)$$

when considering the first derivative between the adjacent model parameters. The matrix \mathbf{R} in eq. 2.67 implies that the difference between the resistivities of two layers must be considered when using the constraint (2.66). In 2D, however, \mathbf{R} consists of two roughening matrices in Z and Y directions (deGroot Hedlin & Constable, 1990). Considering the first derivative, \mathbf{R} is written as,

$$\mathbf{R} = \mathbf{R}_z + \mathbf{W}\mathbf{R}_y = \begin{pmatrix} 1 & -1 & \cdots & 0 \\ 0 & 1 & -1 & \vdots \\ \vdots & \cdots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix} + \mathbf{W} \begin{pmatrix} 1 & 0 & \cdots & 0 & -1 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 & \cdots & 0 & 1 \end{pmatrix} \quad (2.68)$$

where \mathbf{W} has the same structure as \mathbf{R}_y containing the ratio of cell dimension in z and y directions (Δ_z / Δ_y). There are $N_y - 1$ zeroes between the entries in the rows of \mathbf{R}_y .

Denoting \mathbf{m}_n as the model parameter at the n th iteration and \mathbf{J}_n as the sensitivity matrix evaluated at \mathbf{m}_n , the penalty function (eq. 2.64) for $\mathbf{m}_n + \Delta\mathbf{m}$ can be approximated, after linearizing $\mathbf{F}(\mathbf{m})$, as,

$$\begin{aligned} \Psi^n &= \Psi_d^n + \lambda\Psi_m^n \\ &= [\mathbf{d}'_n - \mathbf{J}_n\Delta\mathbf{m}]^T[\mathbf{d}'_n - \mathbf{J}_n\Delta\mathbf{m}] + \lambda[\mathbf{m}_n + \Delta\mathbf{m}]^T\mathbf{R}^T\mathbf{R}[\mathbf{m}_n + \Delta\mathbf{m}] \end{aligned} \quad (2.69)$$

Differentiating eq. 2.69 with respect to $\Delta\mathbf{m}$ and setting the result equal to zero leads to the usual $M \times M$ system of normal equations for a stationary point,

$$\Delta\mathbf{m} = (\mathbf{J}_n^T\mathbf{J}_n + \lambda\mathbf{R}^T\mathbf{R})^{-1}\mathbf{J}_n^T\mathbf{d}'_n - \lambda\mathbf{R}^T\mathbf{R}\mathbf{m}_n \quad (2.70)$$

The penalty function in eq. 2.69 is approximated for the model of the next iteration ($n+1$), since $\mathbf{m}_{n+1} = \mathbf{m}_n + \Delta\mathbf{m}$. However, by approximating the penalty function for the current model (\mathbf{m}_n), eq. 2.70 can be rewritten as (Constable *et al.*, 1987; Egbert, 2006),

$$\mathbf{m}_{n+1} = (\mathbf{J}_n^T\mathbf{J}_n + \lambda\mathbf{R}^T\mathbf{R})^{-1}\mathbf{J}_n^T\hat{\mathbf{d}}_n \quad (2.71)$$

where,

$$\hat{\mathbf{d}}_n = \mathbf{d} - \mathbf{F}(\mathbf{m}_n) + \mathbf{J}_n\mathbf{m}_n$$

Although, solving eq. 2.70 for the model update $\Delta \mathbf{m}$ and adding the result to \mathbf{m}_n is exactly equivalent to solving eq. 2.71 directly for \mathbf{m}_{n+1} .

Before proceeding, a few comments on the regularization term in eq. 2.69 (the second term on the right hand side) are appropriate. The roughness operator $\mathbf{R}^T \mathbf{R}$ can be formulated in terms of a smoothing operator \mathbf{C}_m by setting $\mathbf{C}_m^{-1} = \mathbf{R}^T \mathbf{R}$ (Menke, 1984; Egbert, 2006). Hence, eq. 2.71 can be rewritten using \mathbf{C}_m^{-1} as,

$$\mathbf{m} = (\mathbf{J}^T \mathbf{J} + \lambda \mathbf{C}_m^{-1})^{-1} \mathbf{J}^T \hat{\mathbf{d}} \quad (2.72)$$

\mathbf{C}_m is defined as the smoothing operator which is interpreted as the covariance of the model parameters. This operator provides a priori constraints on the magnitude and spatial smoothness of conductivity variations (Egbert, 2006). The roughness operator is usually used in most model space inversion formulations. As \mathbf{R} is sparse (see i.e. eq. 2.68) the multiplications $\mathbf{R}^T \mathbf{R} \mathbf{m}$ and $\mathbf{R}^T \mathbf{R}$ in eqs. 2.70 and 2.71 are trivial and can be implemented very efficiently. In contrast, formulating the regularized inverse problem in data space using the smoothing operator \mathbf{C}_m (as we will see later) is more appropriate in terms of multiplications (Egbert, 2006).

2.6.2 Sensitivity matrix computation

All inversion schemes presented so far require an explicit expression of the sensitivity matrix (\mathbf{J}) to formulate the normal equations (Gauss-Newton and Occam Inversion). The Non-Linear Conjugate Gradients (NLCG) scheme, which will be discussed later, requires an implicit expression to formulate the matrix-vector products $\mathbf{J}^T \mathbf{d}$.

The definition of the sensitivity matrix suggests that the computation of each element of \mathbf{J} can be done in straightforward manner: by simply perturbing each model parameter by δm , solving the forward modelling problem for $m + \delta m$ and computing the first derivative, yields one column in \mathbf{J} ,

$$\frac{\partial F_l(m_k)}{\partial m_k} = \frac{F_l(m_k) - F_l(m_k + \Delta m)}{\Delta m} \quad l = 1, \dots, N \quad \text{and} \quad k = 1, \dots, M \quad (2.73)$$

These steps must be repeated for all model parameters. This means, that there are M forward modelling computations to get all columns of \mathbf{J} . For a 1D inversion this is not a problem because we have only a few model parameters and data to consider. Furthermore, since analytic solutions exist for the forward modelling problem in 1D, there exists also an analytic solution to compute all elements of \mathbf{J} (Meju, 1994). However, in 2D and 3D cases with hundred of thousands of model parameters, solving the forward modelling problem M times is not efficient.

Rodi (1976) and later Rodi & Mackie (2001) suggested an efficient method to compute all elements of \mathbf{J} with only N forward modelling computations. This method is based on two main concepts:

- Any perturbation of the model parameters causes a perturbation in the electromagnetic field and consequently in the model responses $\mathbf{F}(\mathbf{m})$.
- The reciprocal property of the forward modelling operator: The reciprocity principle states that the model response of a receiver at location j depending on a transmitter located at the location i is similar to that of a transmitter located at the location j and a receiver at i .

To solve the forward modelling problem in 2D and 3D, the linear equations system:

$$\mathbf{A}(\mathbf{m}, \omega)\mathbf{x}(\mathbf{m}, \omega) = \mathbf{b}(\mathbf{m}, \omega) \quad (2.74)$$

is solved for an unknown vector \mathbf{x} . Matrix \mathbf{A} contains discrete representations of gradient (∇), divergence ($\nabla \cdot$) and rotation ($\nabla \times$) of the electromagnetic fields in the earth model at particular points of the grid. The vector \mathbf{b} contains the boundary conditions and a source term. For the 2D case in MT, the source term is the electric field arising from a layered earth model (1D). For the 3D case, the source terms are the electric and magnetic fields arising from 2D structures at the boundaries of the 3D model. The vector \mathbf{x} in eq. 2.74 is what we are interested in. In the 2D case, this vector contains the unknown scalar electric (E-polarisation) or magnetic (B-polarisation) field values defined at the nodes of the 2D grid. In the 3D case, however, the vector \mathbf{x} contains the three components of the electric field arising from electric current system flows in X direction (Ex-Hy polarisation, first mode) and three components of the electric field arising from electric current system flows in Y direction (Ey-Hx polarisation, second mode). Each of these components is defined at the edges of the cubes of a staggered 3D grid. From now on the abbreviation \mathbf{e} will be used instead of \mathbf{x} to represent the electromagnetic solution in eq. 2.74.

To derive a generic expression for computing the elements of the sensitivity matrix in 2D and 3D, the forward modelling operator must be rewritten as a function depending on the solution of eq. 2.74, and possibly but not necessary, on the model parameters \mathbf{m} :

$$\mathbf{F}(\mathbf{m}) = \psi(\mathbf{e}(\mathbf{m}), \mathbf{m}) \quad (2.75)$$

Equation 2.75 states that the model responses are derived from the electromagnetic solution \mathbf{e} . The function ψ can be considered as a function that extracts (via interpolation and other mathematical basic functions) the corresponding values of electric and magnetic fields at the observation location to evaluate the model responses. Rewriting the definition of \mathbf{J} using equation 2.75, yields:

$$\begin{aligned} \mathbf{J} &= \frac{\partial \mathbf{F}(\mathbf{m})}{\partial \mathbf{m}} = \frac{\partial \psi(\mathbf{e}(\mathbf{m}), \mathbf{m})}{\partial \mathbf{m}} \\ &= \frac{\partial \psi}{\partial \mathbf{e}} \frac{\partial \mathbf{e}}{\partial \mathbf{m}} + \frac{\partial \psi}{\partial \mathbf{m}} \end{aligned} \quad (2.76)$$

Writing 2.76 in matrix notation, gives:

$$\mathbf{J} = \mathbf{L}\mathbf{F} + \mathbf{Q} \quad (2.77)$$

where². :

$$\mathbf{L} = \frac{\partial \psi}{\partial \mathbf{e}}, \quad \mathbf{F} = \frac{\partial \mathbf{e}}{\partial \mathbf{m}}, \quad \mathbf{Q} = \frac{\partial \psi}{\partial \mathbf{m}}$$

Matrix \mathbf{L} connects perturbations in the electromagnetic solution to perturbations in the model responses. \mathbf{Q} is used only if the function ψ depends on the model parameters. In general, the matrices \mathbf{L} and \mathbf{Q} represent linearizations of the data functional. Computations of \mathbf{L} and \mathbf{Q} are in general very efficient, since both matrices are sparse and depend only on a few nodes surrounding the corresponding MT station or the model parameter under consideration, respectively (Egbert, 2006). The only computational challenge involved is in computing the elements of \mathbf{F} .

To derive an expression for calculating the elements of \mathbf{F} , we first take the derivative of 2.74 with respect to the model parameters \mathbf{m} . Furthermore, let us assume that \mathbf{b} does not depend on the model parameters, hence, $\frac{\partial \mathbf{b}}{\partial \mathbf{m}} = 0$, we obtain:

$$\begin{aligned} \frac{\partial \mathbf{A}}{\partial \mathbf{m}} \mathbf{e} + \mathbf{A} \frac{\partial \mathbf{e}}{\partial \mathbf{m}} &= 0 \\ \mathbf{A} \frac{\partial \mathbf{e}}{\partial \mathbf{m}} &= -\frac{\partial \mathbf{A}}{\partial \mathbf{m}} \mathbf{e} \\ \mathbf{A}\mathbf{F} &= \mathbf{P} \\ \Rightarrow \mathbf{F} &= \mathbf{A}^{-1}\mathbf{P} \end{aligned} \quad (2.78)$$

where,

$$\mathbf{P} = -\frac{\partial \mathbf{A}}{\partial \mathbf{m}} \mathbf{e}$$

Substituting 2.78 in 2.77 results in a new expression of \mathbf{J} :

$$\mathbf{J} = \mathbf{L}\mathbf{A}^{-1}\mathbf{P} + \mathbf{Q} \quad (2.79)$$

The term $\mathbf{A}^{-1}\mathbf{P}$ in eqs. 2.78 and 2.79 is usually called “pseudo forward modelling”, since solving 2.78 for \mathbf{F} is similar to solving eq. 2.74 for \mathbf{e} . However, the right hand side (sources in eq. 2.74 and fictitious sources in eq. 2.78) differ between both equations. The right hand side in eq. 2.78 contains a matrix with dimension $M \times M$ (the matrix \mathbf{P}). This matrix contains in each column the derivative of the coefficients matrix \mathbf{A} with respect to one model parameter (m_k) multiplied with the values of the EM field surrounding m_k . Since $\frac{\partial \mathbf{A}}{\partial \mathbf{m}}$ is a sparse matrix and supported only for a few rows corresponding to one particular model parameter, \mathbf{P} is also a sparse matrix. Therefore, computing \mathbf{P} requires minimal computational effort.

One way to compute all entries of \mathbf{J} using eq. 2.79 is to solve eq. 2.78 for \mathbf{F} (the pseudo

²Please keep in mind that \mathbf{F} in eq. 2.77 is unrelated to the forward modelling operator $\mathbf{F}(\mathbf{m})$.

forward modelling) $M \times NPer$ times, where $NPer$ is the total number of periods³ and then evaluate eq. 2.79 (Rodi, 1976; Rodi & Mackie, 2001; Egbert, 2006). Furthermore, if the data vector contains data from more than one polarisation (i.e. TE and TM in 2D case) then the number of the pseudo forward modelling problems to be solved is $M \times NPer \times NPol$, where $NPol$ is the number of polarisations.

Taking into account the reciprocity principle, by simply taking the transpose of eq. 2.79 and noting that \mathbf{A} is symmetric, we obtain,

$$\mathbf{J}^T = \mathbf{P}^T \mathbf{A}^{-1} \mathbf{L}^T + \mathbf{Q}^T \quad (2.80)$$

In this case, the sensitivity matrix can in fact be obtained by solving the pseudo forward modelling problem N times instead of $M \times NPer$ times (Rodi & Mackie, 2001; Egbert, 2006). This is the usual reciprocity application for efficient calculation of \mathbf{J} . This means, instead of setting the sources in each model parameter to solve 2.78 for \mathbf{F} , we can set the sources at the observation locations and solve $\mathbf{A}\mathbf{F} = \mathbf{L}^T$ for \mathbf{F} (as reciprocity principle holds). An example in appendix A demonstrates the mathematical meaning of the sensitivity matrix computation in the 2D MT case.

In the 3D MT case, the impedance is a 2×2 tensor, which requires two independent electric field solutions computed for two different source polarisations (Weaver, 1994; Egbert, 2006; Siripunvaraporn *et al.*, 2005):

$$\begin{aligned} \mathbf{Z} &= \mathbf{E}\mathbf{B}^{-1} \\ \begin{pmatrix} Z_{xx} & Z_{xy} \\ Z_{yx} & Z_{yy} \end{pmatrix} &= \begin{pmatrix} E_x^1 & E_x^2 \\ E_y^1 & E_y^2 \end{pmatrix} \begin{pmatrix} B_x^1 & B_x^2 \\ B_y^1 & B_y^2 \end{pmatrix}^{-1}, \end{aligned} \quad (2.81)$$

The indices 1 and 2 are used to distinguish between solutions for two different source polarisation. Using Kramer's rule to compute the inverse of \mathbf{B} , the components of the impedance tensor can be rewritten as (Newman & Alumbaugh, 2000):

$$\begin{aligned} Z_{xx} &= \frac{E_x^1 B_y^2 - E_x^2 B_y^1}{D}, & Z_{xy} &= \frac{E_x^2 B_x^1 - E_x^1 B_x^2}{D} \\ Z_{yx} &= \frac{E_y^1 B_y^2 - E_y^2 B_y^1}{D}, & Z_{yy} &= \frac{E_y^2 B_x^1 - E_y^1 B_x^2}{D} \end{aligned} \quad (2.82)$$

where,

$$D = B_x^1 B_y^2 - B_x^2 B_y^1$$

The derivative of \mathbf{Z} with respect to \mathbf{m} ($\partial_{\mathbf{m}}\mathbf{Z}$) can be found when considering that perturbations to the computed EM fields ($\partial_{\mathbf{m}}\mathbf{E}$, $\partial_{\mathbf{m}}\mathbf{B}$) result in first order perturbation of the impedance tensor (Egbert, 2006):

$$\begin{aligned} \frac{\partial \mathbf{Z}}{\partial \mathbf{m}} &= \frac{\partial}{\partial \mathbf{m}} \left(\frac{\mathbf{E}}{\mathbf{B}} \right) \\ &= \frac{\partial \mathbf{E}}{\partial \mathbf{m}} \mathbf{B}^{-1} - \mathbf{Z} \frac{\partial \mathbf{B}}{\partial \mathbf{m}} \mathbf{B}^{-1} \end{aligned} \quad (2.83)$$

³Because \mathbf{P} has M columns and the values in each column are different for each period.

For 3D forward modelling computations, the equations system $\mathbf{A}_i \mathbf{e}_i = \mathbf{b}_i$, where i denotes the two source polarisations ($i=1,2$), is solved for the electric field \mathbf{E} . The magnetic field can then be numerically determined from Faraday's law:

$$\mathbf{B} = \nabla \times \mathbf{E}/(-i\omega) \quad (2.84)$$

by approximating the curl of the electric field at various corner points of the model cubes (Alumbaugh *et al.*, 1996; Weaver, 1994). The calculated electric field and the approximated magnetic field can then be interpolated to the station location on (or in) the 3D grid. This means, that for one particular station, we can write:

$$\begin{aligned} E_x^1 &= \lambda_{Ex}^1 \mathbf{e}_1 & E_x^2 &= \lambda_{Ex}^2 \mathbf{e}_2 & E_y^1 &= \lambda_{Ey}^1 \mathbf{e}_1 & E_y^2 &= \lambda_{Ey}^2 \mathbf{e}_2 \\ B_x^1 &= \lambda_{Bx}^1 \mathbf{e}_1 & B_x^2 &= \lambda_{Bx}^2 \mathbf{e}_2 & B_y^1 &= \lambda_{By}^1 \mathbf{e}_1 & B_y^2 &= \lambda_{By}^2 \mathbf{e}_2 \end{aligned} \quad (2.85)$$

As in the 2D case, the basic functions λ_{Ex} and λ_{Ey} interpolate the calculated electric fields at the edges of the 3D grid cubes, and then interpolate to the point of interest (i.e. station location). The functions λ_{Bx} and λ_{By} must also approximate $\nabla \times \mathbf{E}$ to compute \mathbf{B} at all faces of the cubes in the 3D grid.

We turn our attention to solve the pseudo forward modelling problem $\mathbf{A}^{-1} \mathbf{L}^T$, for which we must first compute the rows of \mathbf{L} . Since perturbation of the EM field solutions causes perturbations to \mathbf{Z} , we may write:

$$\frac{\partial \mathbf{Z}}{\partial \mathbf{m}} = \mathbf{L} \quad (2.86)$$

Writing eq. 2.83 in its components and using eqs. 2.85, we can write a general form of \mathbf{L} as:

$$\partial_m Z_{ij} = \sum_{k=1}^2 \left\{ (B_j^k)^{-1} [\lambda_E^i - Z_{i1} \lambda_B^1 - Z_{i2} \lambda_B^2] \right\} \partial_m e_k \quad (2.87)$$

where $i, j = x, y$.

Finally, let us consider the computation of \mathbf{Q} . The matrix \mathbf{Q} is computed by taking the derivative of $\psi(\mathbf{e}, \mathbf{m})$ with respect to \mathbf{m} . In order to know if \mathbf{Q} is needed in eqs. 2.79 and 2.80 or not, we have to investigate the dependency of ψ on \mathbf{m} . As discussed above, the function ψ takes the computed electric fields \mathbf{e} and the model parameter as an input to compute the required EM field components. In the 2D E-polarisation case (TE), we compute the x component of the electric field (E_x). In order to compute the horizontal transfer function Z_{xy} , we need the horizontal orthogonal magnetic field component B_y , since $Z_{xy} = \frac{E_x}{B_y}$. However, B_y can be computed from E_x by taking the curl of \mathbf{E} and noting that $E_y = E_z = 0$ in the 2D E-polarization case (see eq. 2.36c). This means,

$$B_y = \frac{\partial E_x}{\partial z} \left(\frac{1}{-i\omega} \right) \quad (2.88)$$

From eq. 2.88 we can deduce that computing B_y from E_x does not depend on \mathbf{m} . Consequently, ψ is independent of \mathbf{m} , therefore,

$$\mathbf{Q} = \frac{\partial \psi}{\partial \mathbf{m}} = 0$$

In the 2D B-polarization case (TM), we compute the y component of the magnetic field (B_y). To compute the horizontal transfer function Z_{yx} , we need the horizontal electric field component E_y , since $Z_{yx} = \frac{E_y}{B_x}$. As before, E_y is computed from B_y using (see eq. 2.37b),

$$E_y = -\frac{\partial B_x}{\partial z} \left(\frac{1}{\mu_0 \sigma} \right) \quad (2.89)$$

Obviously, computing E_y from B_x depends on \mathbf{m} (σ in eq. 2.89), therefore,

$$\mathbf{Q} = \frac{\partial \psi}{\partial \mathbf{m}} \neq 0$$

In the 3D case, we have a similar situation as in the 2D TE case. We usually solve the forward modelling problem for the electric field and the magnetic field is determined from the electric field (see 2.84), in which there is no dependency on \mathbf{m} . Thus,

$$\mathbf{Q} = \frac{\partial \psi}{\partial \mathbf{m}} = 0$$

The procedures presented above to compute the sensitivity matrix in both 2D and 3D cases have very useful properties that can be used later in parallelizing their computations:

- The columns of \mathbf{L} and the solutions of the pseudo forward modelling problem $\mathbf{A}^{-1}\mathbf{L}^T$ can be computed independently for all periods. This means that we can distribute this computational task among several processors.
- $\mathbf{A}^{-1}\mathbf{L}^T$ is solved using only one column in \mathbf{L}^T . Consequently, one row of \mathbf{J} (corresponding to one station) does not depend on the computation of other rows in \mathbf{J} .

2.6.3 Solution of the normal equations

The normal equations. (i.e. eqs. 2.65, 2.71 and 2.72) presented above, can be rewritten in matrix vector notation as:

$$\mathbf{H}\mathbf{x} = \mathbf{g} \quad (2.90)$$

where,

$$\mathbf{H} = \mathbf{J}^T \mathbf{J} + \begin{Bmatrix} \beta \mathbf{I} \\ \lambda \mathbf{R}^T \mathbf{R} \\ \lambda \mathbf{C}_m^{-1} \end{Bmatrix}, \quad \mathbf{g} = \mathbf{J}^T \begin{Bmatrix} \mathbf{d}' \\ \mathbf{d}' - \lambda \mathbf{R}^T \mathbf{R} \mathbf{m} \\ \hat{\mathbf{d}} \end{Bmatrix}, \quad \mathbf{x} = \begin{Bmatrix} \Delta \mathbf{m} \\ \Delta \mathbf{m} \\ \mathbf{m} \end{Bmatrix}$$

The real symmetric positive definite equation system presented in eq. 2.90 can be solved by applying a standard equation system solver. Equation system solvers can be classified in two main groups, direct and iterative solvers. Examples for the direct solvers are the Cholesky and LU decompositions. Both methods decompose the matrix \mathbf{H} in a triangular matrix \mathbf{U} (Cholesky decomposition), or in two triangular matrices \mathbf{L} and \mathbf{U} (LU decomposition). The resulting triangular matrices must satisfy the condition,

$$\mathbf{H} = \mathbf{U}^T \mathbf{U} \quad \mathbf{H} = \mathbf{L} \mathbf{U}$$

By forward substitution,

$$\mathbf{U}^T \mathbf{y} = \mathbf{g} \quad \text{or} \quad \mathbf{U} \mathbf{y} = \mathbf{g}$$

and backward substitution,

$$\mathbf{U} \mathbf{x} = \mathbf{y} \quad \text{or} \quad \mathbf{L} \mathbf{x} = \mathbf{y}$$

a solution for vector \mathbf{x} which correspond to $\Delta \mathbf{m}$ or \mathbf{m} can be found.

An example for an iterative solver for the inversion problem is the conjugate gradient method (CG). To solve the system of equations (eq. 2.90), CG generates a sequence of approximated solutions by iterative minimisation of the quadratic form:

$$\Phi(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} - \mathbf{g}^T \mathbf{x} \quad (2.91)$$

along a sequence of conjugate search directions \mathbf{p} (Press et al., 1986). Starting from $\mathbf{p}_0 = 0$ as a first guess, the next search direction is obtained using:

$$\mathbf{p}_{k+1} = \mathbf{h}_{k+1} - \gamma_k \mathbf{p}_k \quad (2.92)$$

where \mathbf{h}_{k+1} is the gradient of eq. 2.91 at the solution \mathbf{x}_k and given by:

$$\mathbf{h}_{k+1} = \frac{\partial}{\partial \mathbf{x}_k} \left(\frac{1}{2} \mathbf{x}_k^T \mathbf{H} \mathbf{x}_k - \mathbf{g}^T \mathbf{x}_k \right) = \mathbf{H} \mathbf{x}_k - \mathbf{g} \quad (2.93)$$

The scalar γ_k in equation 2.92 ensures that the search direction \mathbf{p}_k is conjugate to all other previous search directions, which means:

$$\mathbf{p}_k^T \mathbf{H} \mathbf{p}_{k'} = 0 \quad k' < k$$

The scalar γ_k is calculated as:

$$\gamma_k = \frac{\mathbf{h}_{k+1} \mathbf{H} \mathbf{p}_k}{\mathbf{p}_k^T \mathbf{H} \mathbf{p}_k} \quad (2.94)$$

The solution \mathbf{x}_{k+1} which ensures that $\Phi(\mathbf{x})$ in its minimum is obtained using the iteratively line search procedure:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \beta_{k+1} \mathbf{p}_{k+1} \quad (2.95)$$

where the scalar step size β is given by:

$$\beta_{k+1} = \frac{\mathbf{p}_{k+1}^T \mathbf{h}_{k+1}}{\mathbf{p}_{k+1}^T \mathbf{H} \mathbf{p}_{k+1}} \quad (2.96)$$

In the CG scheme the main computation tasks are the matrix vector multiplications $\mathbf{H}\mathbf{p}$ required in eqs. 2.96 and 2.94. Rodi & Mackie (2001) showed that this multiplication can be accomplished at the cost of two pseudo forward modelling computations⁴. Moreover, even computing and storing the sensitivity matrix is not required. Let \mathbf{f} denote the matrix vector product,

$$\mathbf{f} = \mathbf{J}\mathbf{p} \quad (2.97)$$

The matrix vector product $\mathbf{H}\mathbf{p}$ in connection with one of the normal equations (i.e Occam inversion eq. 2.71) can be rewritten as:

$$\begin{aligned} \mathbf{H}\mathbf{p} &= (\mathbf{J}^T \mathbf{J} + \lambda \mathbf{R}^T \mathbf{R})\mathbf{p} \\ &= \mathbf{J}^T \mathbf{J}\mathbf{p} + \lambda \mathbf{R}^T \mathbf{R}\mathbf{p} \\ &= \mathbf{J}^T \mathbf{f} + \lambda \mathbf{R}^T \mathbf{R}\mathbf{p} \end{aligned} \quad (2.98)$$

In scope of eqs. 2.79 and 2.80 the matrix vector products $\mathbf{J}\mathbf{p}$ and $\mathbf{J}^T \mathbf{f}$ can be calculated as,

$$\mathbf{J}\mathbf{p} = \mathbf{L}\mathbf{A}^{-1}\mathbf{P}\mathbf{p} + \mathbf{Q}\mathbf{p} \quad (2.99)$$

and

$$\mathbf{J}^T \mathbf{f} = \mathbf{P}^T \mathbf{A}^{-1} \mathbf{L}^T \mathbf{f} + \mathbf{Q}^T \mathbf{f} \quad (2.100)$$

The terms $\mathbf{A}^{-1}\mathbf{P}\mathbf{p}$ and $\mathbf{A}^{-1}\mathbf{L}^T \mathbf{f}$ are the two additional pseudo forward modelling solutions required in CG to avoid an implicit computation of the sensitivity matrix. The CG scheme is usually implemented within an outer iterative loop that minimises the objective function Ψ in eq. 2.64. The flow chart of the algorithm to solve the normal equations (i.e. eq. 2.71) with CG could look as in fig.2.9:

2.6.4 Non linear inversion using conjugate gradient (NLCG)

Nonlinear conjugate gradients (NLCG) is closely related to the linear CG method described above in the context of solving the linearized inverse problem. However, the NLCG method solves directly minimisation problems that are not quadratic and avoid the iterative linearized inversion procedure used in Gauss-Newton style inversion (GN) (Rodi & Mackie, 2001).

NLCG have been used to solve the inverse problem in 2D MT (Rodi & Mackie, 2001) and in 3D MT (Newman & Alumbaugh, 2000). Numerical tests showed that the NLCG is more efficient than GN in terms of both computer memory requirements

⁴Note that a forward modelling requires computation of all periods and modes.

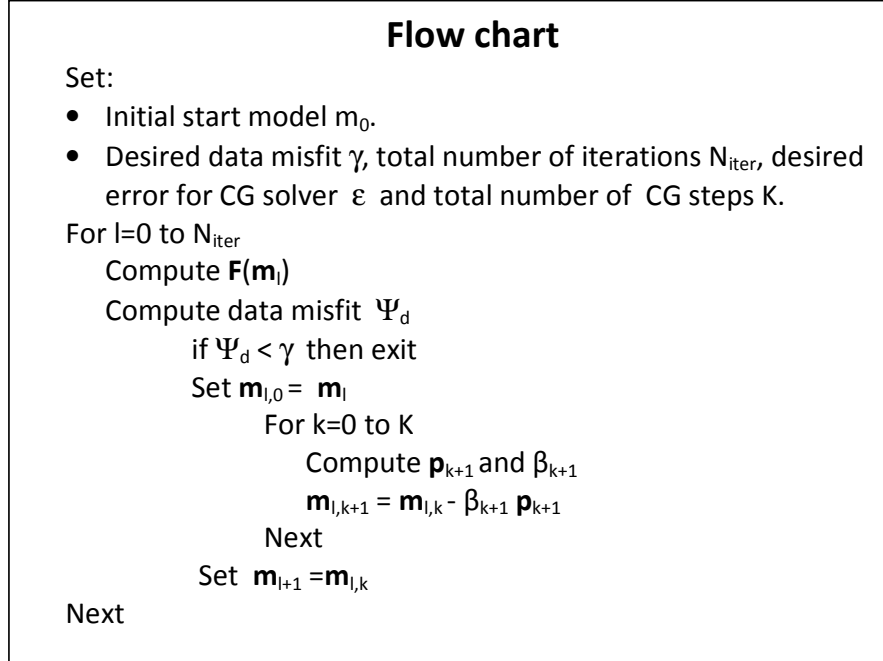


Figure 2.9: The flow chart for the conjugate gradient scheme in solving the linearized normal equation.

and CPU time needed to solve the problem (Rodi & Mackie, 2001). For large scale problems, as in the 3D case, inversion based on GN algorithms may not be feasible due to the fact that in these algorithms the sensitivity matrix (\mathbf{J}) must exist and the cross product $\mathbf{J}^T \mathbf{J}$ must be calculated to solve the normal equation. Because NLCG avoids explicit use of \mathbf{J} and consequently, the cross product $\mathbf{J}^T \mathbf{J}$, it seems to be a realistic option to solve the inverse problem in the 3D MT case. However, in comparison with GN, the NLCG requires more CG steps for convergence. This is because the GN exhibits quadratic convergence and requires only few iterations (the outer loop in fig. 2.9) required for convergence (Egbert, 2006).

NLCG utilises essentially the same basic computational steps as required for solving the linearized problem using a CG approach. With the NLCG approach one must evaluate the gradient of eq. 2.64 with respect to the model parameters \mathbf{m} . However, one must take into account the non linearized form of Ψ_d (eq. 2.55). In this case we can write:

$$\nabla \Psi = \nabla \Psi_d + \lambda \nabla \Psi_m \quad (2.101)$$

Evaluation of the gradient of the model smoothness function Ψ_m presented in eq. 2.66 leads to:

$$\nabla \Psi_m = 2\mathbf{R}^T \mathbf{R} \mathbf{m} \quad (2.102)$$

The gradient of eq. 2.55 is given by:

$$\nabla \Psi_d = -2\mathbf{J}^T (\mathbf{d} - \mathbf{f}(\mathbf{m})) \quad (2.103)$$

The main objective with the NLCG approach is to minimise the penalty function Ψ , however, with respect to β (the step size in eq. 2.95).

From eqs. 2.102 and 2.102 we can deduce that the main computational task is the matrix vector product $\mathbf{J}^T(\mathbf{d} - \mathbf{f}(\mathbf{m}))$. However, as discussed above, \mathbf{J} must not exist to accomplish this multiplication.

2.6.5 Linearized inversion in data space

The solution of the normal equations in model space (i.e. eq. 2.90) requires solving an equation system which has a dimension equal to $M \times M$, where M is the number of model parameters. The difficulty in solving such equation systems becomes clear when solving the 3D inversion problem, since in 3D inversion M can be as large as hundreds of thousands while the number of data points N is very small ($M \gg N$). Therefore, it is more appropriate to convert the inverse problem from model space to data space and solve an equation system which has a dimension equal to $N \times N$.

Let us consider the Occam inversion scheme discussed above, however using the smoothing \mathbf{C}_m rather than the roughness operator $\mathbf{R}^T\mathbf{R}$ (eq. 2.72). Siripunvaraporn *et al.* (2005) show that the solution of eq. 2.72 for the model parameters \mathbf{m}_{k+1} at iteration $k+1$, can be expressed using the sensitivity matrix smoothed by \mathbf{C}_m . Using the properties $(\mathbf{J}^{-T})^{-1} = \mathbf{J}^T$ and the matrix notation $(\mathbf{A}\mathbf{B})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$, eq. 2.72 can be rewritten as⁵:

$$\mathbf{m}_{k+1} - \mathbf{m}_k = (\mathbf{J}_k^T \mathbf{C}_d^{-1} \mathbf{J}_k + \lambda \mathbf{C}_m^{-1})^{-1} \mathbf{J}_k^T \mathbf{C}_d^{-1} \hat{\mathbf{d}}_k \quad (2.104)$$

$$= \mathbf{C}_m \mathbf{J}_k^T \boldsymbol{\beta}_{k+1} \quad (2.105)$$

where,

$$\boldsymbol{\beta}_{k+1} = (\mathbf{J}_k \mathbf{C}_m \mathbf{J}_k^T + \lambda \mathbf{C}_d)^{-1} \hat{\mathbf{d}}_k \quad (2.106)$$

is an unknown expansion coefficient vector of the basis function $\mathbf{C}_m \mathbf{J}_k$ (Siripunvaraporn *et al.*, 2005; Siripunvaraporn & Egbert, 2000; Egbert, 2006).

With the transformation from model space (eq. 2.104) to data space (eq. 2.105) two main advantages are achieved:

- The dimension of the problem to be solved for the model parameter vector \mathbf{m}_{k+1} has been reduced from $M \times M$ to $N \times N$. Because computing the \mathbf{m}_{k+1} requires first solving 2.106 for the vector $\boldsymbol{\beta}_{k+1}$ and then substituting in eq. 2.105.
- The inverse of the smoothing matrix \mathbf{C}_m in eq. 2.104 is not required any more in eq. 2.105.

⁵The data covariance matrix \mathbf{C}_d has been used explicitly here to be consistent with the published papers on this subject

“The solutions obtained from the Occam inversion in the model and the data space, should in theory be identical if all parameters used are the same” (Siripunvaraporn *et al.* , 2005)

Nevertheless, for the Occam inversion in the data space we still need to compute the sensitivity matrix (\mathbf{J}) and the cross product $\mathbf{J}\mathbf{J}^T$. These two computational efforts are the most time and memory consuming when running the 3D inversion of MT data. Furthermore, several computation processes, i.e multiplying \mathbf{J} by \mathbf{C}_m or by the solution vector $\boldsymbol{\beta}$ must be accomplished with \mathbf{J} . This means that \mathbf{J} must exist as a whole matrix in computer memory.

Considering the properties of the sensitivity matrix computation discussed in 2.6.2 and decomposing the computing operations involved in eqs. 2.106 and 2.105, very efficient scheme, using parallelization, can be found to minimise both the time and memory requirements to solve the inverse problem in 3D MT. These issues will be discussed in detail in the next chapter.

Chapter 3

A generic, efficient and practical parallelization method for MT forward modelling and inversion in 2D and 3D

3.1 Introduction

Modern natural sciences (e.g. geosciences, bioscience, chemistry, and physics) depend, among others, on computer science and the developments of the computer industry. Today, it is inconceivable to simulate various four-dimensional (4D) real world problems (e.g. climate changes or geodynamic processes to explain plate tectonics) or even to solve a set of complex equations, without using computing resources. However, solving such kind of problems on a personal computer can easily exceed the limits of existing hardware (memory space and CPU) due to the large amount of both code instructions and number of data required to solve the problem. Executing the code (computer instructions) in a serial manner can take very long time. An apparently straightforward way to overcome this problem is to connect several powerful computer machines and let them work together to solve simultaneously (in parallel) a particular numerical problem.

The concept of parallelization follows simple natural principals: If we are asked to produce n end products (some results) by using m elements (data) and we have only one worker (processor), then the time required to achieve this task takes the time t . But if we employ w workers to achieve the same task, then the time can be reduced ideally to t/w . Furthermore, if we ask all workers to deliver their end product directly to the client (any subroutine that needs these results) then we do not need to have a huge storage (global memory) to keep the end products in it. Instead, small containers (local memory) near each worker will be enough to keep his end product (local

result). With these considerations in mind we can reduce both the time and space required to accomplish the task.

Several powerful computer machines in a network configuration create a so called supercomputer or parallel computer. Parallel computer architectures were already introduced in the sixties of the last century. At that time, parallel computers were defined as collections of several processors (and other hardware components) connected, but not necessarily, to each other by a network to realise one computing task in parallel. Parallel computing has been considered to be the high end of computing, and has been used to model difficult scientific and engineering problems.

For forward modelling and inversion of MT data in 3D, we are facing the same problems with regard to computation time and memory requirements. The approximation of electromagnetic (EM) differential equations on 3D staggered grids can easily exceed the limits of the machine's memory. Furthermore, inverting models of real world problems can also exceed the limits of memory due to large matrices and vectors associated with the inversion process. Even if the required memory is available, the time required to run the inversion process for 3D MT can take several days or months. Thus, adopting parallel computing is inevitable if we want to solve the forward modelling problem and inversion in 3D MT in reasonable time. This is the ultimate ambition of this chapter, which demonstrates strategies and methods to achieve these goals.

The hardware configuration of parallel computers can be classified using Flynn's taxonomy. Flynn (1972) classifies the architectures of parallel computers along two independent parameters: Code instructions and data. Each of these parameters can have only one of two possible states: Single or Multiple. According to this classification, a parallel computer can fall into one of the following categories (Fig. 3.1):

- **Single Instruction, Single Data (SISD):** This kind of computer architecture belongs to a serial (non-parallel) computer. Here, only one single code instruction is executed on one processor at time t using one single data element. This is the oldest and even today, the most common type of computers (Fig. 3.1a).
- **Single Instruction, Multiple Data (SIMD):** A type of parallel computer. With this type of computer architecture, all processors execute one single code instruction at the time t but using different data elements. Most modern computers (e.g. dual or quad cores) belong to this category (Fig. 3.1b).
- **Multiple Instruction, Single Data (MISD):** A single data element fed into multiple processors. Each processor operates on the data element by executing different code instructions. There are not many examples of this computer architecture (Fig. 3.1c).
- **Multiple Instruction, Multiple Data (MIMD):** Here, every processor may

be executing different code instructions and using different data elements. Most of modern parallel computers fall into this category (Fig. 3.1d).

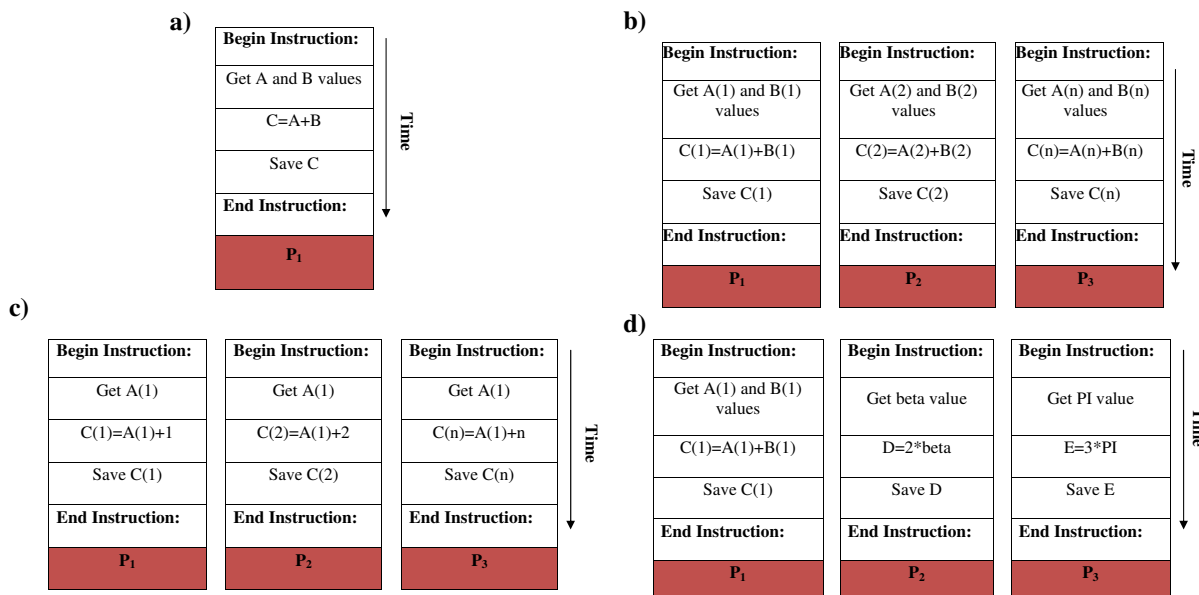


Figure 3.1: Classification of parallel computer architectures after Flynn (1972). a) Single Instruction, Single Data (SISD), b) Single Instruction, Multiple Data (SIMD), c) Multiple Instruction, Single Data (MISD) and d) Multiple Instruction, Multiple Data (MIMD). P_1 , P_2 and P_3 indicate processors IDs.

Classification of parallel computers in terms of instruction execution and data handling lead us to another important question: Where are the data located? Answering this question requires defining and classifying memory architectures of parallel computer. The most important and widely used memory architectures are (Fig. 3.2):

- **Shared Memory.** In this kind of memory, all processors share one global memory and all of them equally have access to it (Fig. 3.2a). This configuration allows individual processors to work independently. Most of modern personal computers consist of several processors (dual or quad) and share one memory (RAM)¹. The main advantage of the shared memory is that the processors do not require a network connection to exchange the data. This is because all data are located in the same memory and any change in the data is immediately visible to other processors. The main disadvantage of this kind of memory is the limitation of the number of processors that share one memory. Since increasing the number of processors increases the traffic on the shared memory, this consequently decreases the computation performance.

¹RAM: Random Access Memory

- **Distributed Memory.** Most bigger parallel computers, which may consist of several hundreds of processors have such memory. As the name indicates, each processor contains its own memory space and only the local processor can change data in the memory (Fig. 3.2b). In this case, data exchange between processors must be realised via network connections, but we can use as many processors as we want without affecting the local traffic to the memory. The disadvantage of this kind of memory is the network connection between the processors. This becomes a problem if we have to distribute computational tasks or data among several hundreds of processors. In this case, the overall performance mainly depends on the speed of the network connection and the amount of data to be transferred. Optimization strategies must be considered to avoid excessive data transfer between processors by using an appropriate parallelization scheme.
- **Hybrid Distributed-Shared Memory.** In this type of computer memory architecture, several (usually dual or quad) processors are attached to one shared memory to create a so called node (or Symmetric Multi Processors, SMP). All SMP nodes can then be connected to each other via network connections (Fig. 3.2c). In this case, the nodes are considered to have distributed memory and within each SMP node they share one memory. Today, the most powerful computers are clusters of this type.

The hardware configurations discussed above will be useless if they are not supported by appropriate software to:

- allow efficient communication: Avoid memory-to-memory copying, allow overlap of computation and communication.
- establish cooperative protocol operations to facilitate data exchange between processors (send and receive),
- synchronise data exchange between processors and run parallel processes.

What is required is a so called communication library. The earliest development in this context are communication libraries like CHIMP (*Common High-level Interface to Message-Passing*) and PUL (*Parallel Utilities Library*) (Bruce *et al.*, 1993) developed at the University of Edinburgh, PVM (*Parallel Virtual Machine*) developed mainly at the University of Tennessee and PICL (*Portable Instrumented Communication Library*) developed at the Oak Ridge National Laboratory. Most of these tools were developed to help programmers solving a particular computation problem in parallel (Kowalik & Grandinetti, 1993).

The first standard of a communication library for parallel computing is called MPI (*Message Passing Interface*), which was developed in the nineties at the University

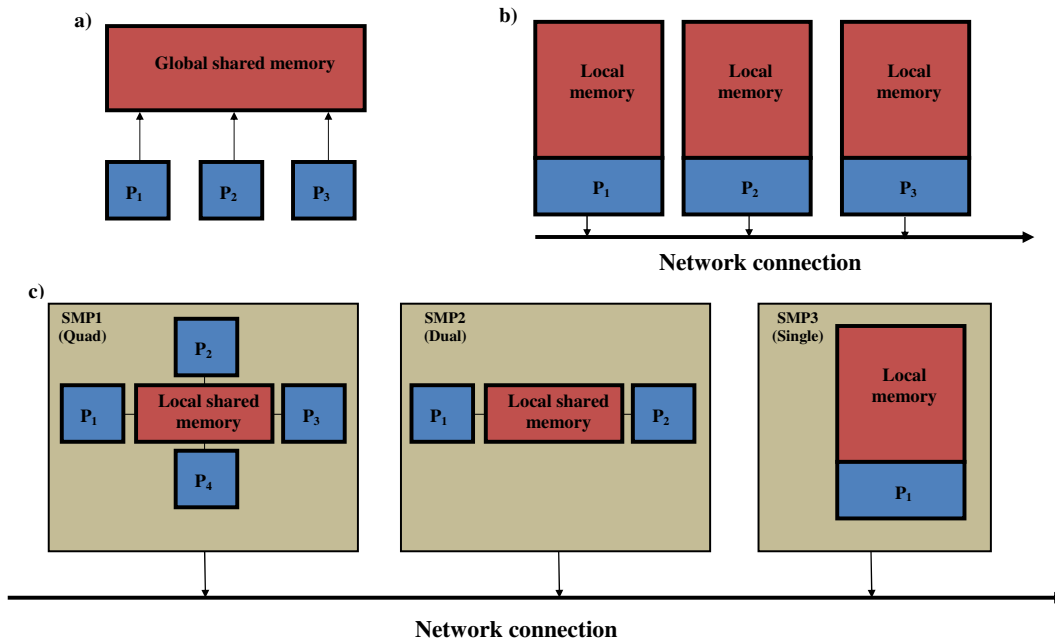


Figure 3.2: Classification of parallel computer memory architectures. a) Three processors (P_1 , P_2 and P_3) sharing one global memory (e.g. triple core machine). b) Each processor owns its own local memory (distributed memory). c) Hybrid distributed shared memory, where each SMP has one local shared memory.

of Tennessee in cooperation with many researchers from universities, government laboratories, and industry from the USA and Europe. The MPI library is a collection of subroutines which can be used in several programming languages (i.e. Fortran 77/90/95, C and C++) to establish effective communication between processors and to ensure data exchange between them. As the name suggests, MPI is based on messages being exchanged between processors. Messages can consist of any data type used usually in programming languages (i.e. in Fortran: Integer, Real,..., etc) and their dimensions can be scalars, vectors or matrices.

In the next few sections, the main concept of MPI along with MPI-subroutines used intensively for this thesis will be introduced to emphasise their contribution to solve the forward modelling and inversion problems in MT.

3.2 Introduction to parallel computing using MPI

MPI addresses a major problem of parallel computing, where data must be moved from the address space of one processor to that of another processor through cooperative operations on each processor. MPI is not a programming language, but instead, all MPI operations are expressed as functions, subroutines, or methods, according to the appropriate language bindings. MPI standard libraries exist for C, C++, Fortran-77,

and Fortran-95. The main goals of MPI, as developers state (Forum, 2008), are:

- Developing a widely used standard for writing message-passing programs. As such, the interface should establish a practical, portable, efficient, and flexible standard for message passing.
- Allow convenient C, C++, Fortran77, and Fortran95 bindings for the interface.
- Allow for implementations that can be used in a heterogeneous environment. This means that parallel programs written by using MPI may run on distributed-memory multiprocessor, networks of workstations, and combinations of all of these. In addition, shared-memory implementations, including those for multi-core processors and hybrid architectures, are possible.
- Define an interface that can be implemented on many vendors platforms (e.g. Linux, Unix and Windows), with no significant changes in the underlying communication and system software.

Sending and receiving of messages by processors is the basic MPI communication mechanism. The communication takes place between processors located in one or different so called communicators. The so called `MPI_COMM_WORLD` is the initially defined universe intra-communicator for all processors. To conduct communications, the MPI library is initialised by calling `MPI_INIT`. From the `MPI_COMM_WORLD` communicator one can create groups of one or more dedicated processors (Fig. 3.3a and b). By definition, the communicator has the following properties:

- A communicator is an opaque object with a number of attributes together with simple rules that govern its creation, use, and destruction.
- Each communicator contains a group of valid processors. The source and destination of a message is identified by processor ID within that group.
- Intra-communication is used for communicating within a single group of processors.
- Inter-communication is used for communicating within two or more groups of processors.
- Communicators are dynamic, i.e., they can be created and destroyed during program execution.

Creating groups of processors from the common world communicator (Fig 3.3b) is useful in many cases. For example, if we intend to broadcast a message for only few processors by using the collective communication `MPI_Bcast` (see below for explanation), then it is possible to create a group containing only the required number of

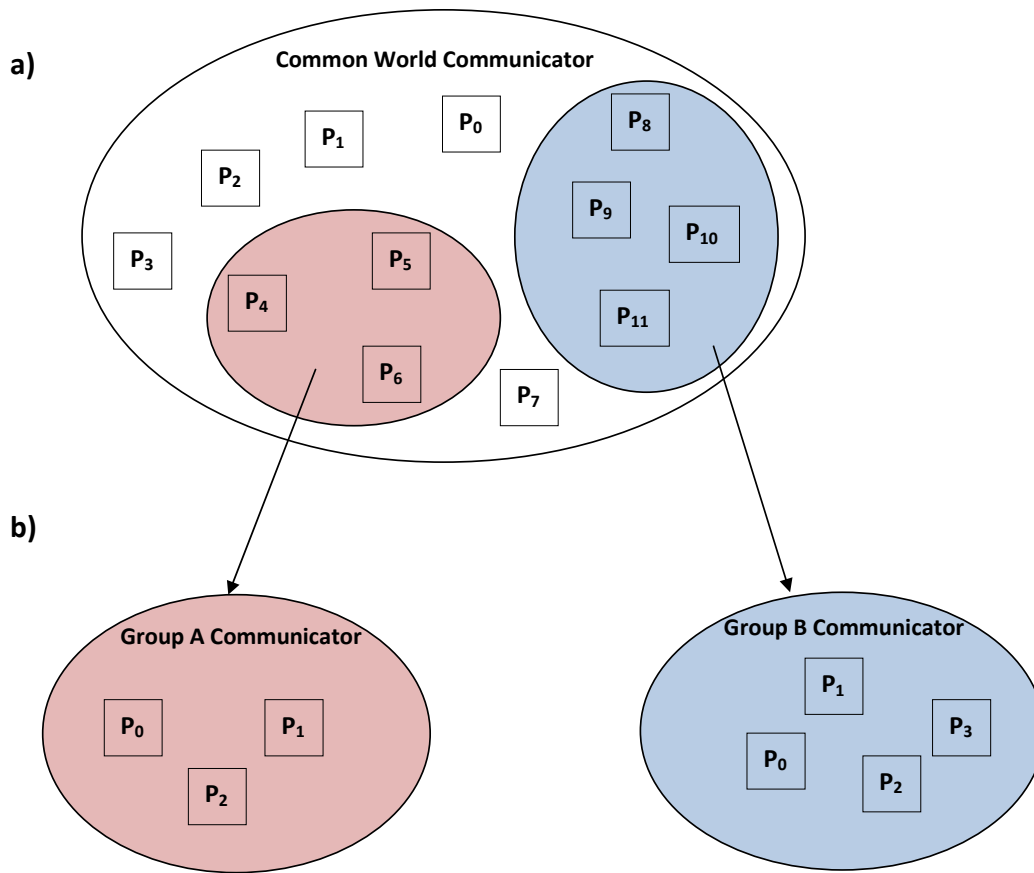


Figure 3.3: a) The base group, upon which all other groups are defined, is the group associated with the initial communicator: The MPI Common World Communicator. It contains all processors that have been initialised when starting MPI. b) Two groups (called A and B) created from the base communicator. Each group contains several processors numbered from 0 to $N_{\text{Proc}}-1$. The communicators A and B have the same properties as the initial common world communicator.

processors. The function `MPI_Bcast` will only affect this group of processors. Furthermore, one can create several groups to distribute various computation tasks among them.

In the MPI communication library mechanism, we can distinguish between three general categories:

- **One-sided:** MPI defines one-sided communication operations that allow one local (active, in the context of sending and receiving messages) processor to directly read from or write to the memory of a remote (passive) processor by using Remote Memory Access (RMA). In this case, only the local processor specifies all communication parameters.
- **Point-to-Point** or two sided: This most basic form of communication in MPI

allows a program to send and receive messages from one specified processor to another specified processor located in the same or a different communicator.

- **Collective communication:** This communication involves all processes in the scope of a communicator. It is the programmer's responsibility to ensure that all processes within a communicator participate in any collective operation by calling the same operation on all processors.

Point-to-Point and Collective communication are the most widely used types of communication when using the MPI library. The developed parallel concepts in this thesis use mainly Point-to-Point and collective communication. Thus, all MPI-subroutines explained later are part of Point-to-Point or collective communication.

Distributing messages between SMP nodes (inter-node connections) is a major issue in MPI and has enormous impact on the computation performance. We can distinguish between several principal types of inter-node connections (Fig. 3.4). Let us assume that we have 4 SMP nodes (with one processor each) associated with the IDs P_0 to P_3 . Let P_0 be the root node or master node. Possible ways to distribute one message to all processors are:

- The master node executes a loop over all processors. In this case, the master enters the loop: First it starts communication with P_1 . P_0 sends the message to P_1 and closes the communication. Afterwards, the same procedure is carried out between P_0 and the other processors (Fig. 3.4a).
- The master node starts communication with P_1 then it sends the message to P_1 , and closes the communication. P_1 , in its turn, starts communication with P_2 , sends the message to P_2 , and finally closes the communication, etc (chain message) (Fig. 3.4b).
- Tree-Structure: The root node starts communication with P_2 , sends the message to P_2 , and closes the communication. Afterwards, P_0 starts communicating with P_1 to send the message. At the same time, P_2 starts communicating with P_3 to send the message (Fig. 3.4c).

The tree-structure is the most efficient way to distribute a message, because it reduces the number of nodes waiting for a message. However, unless we know the underlying topology of the used cluster, we cannot easily decide which scheme to adopt. Ideally, we would like to use a simple function customised to a particular machine, ensuring that we do not have to worry about tedious details, and that we do not have to modify our code every time we change machines.

Fortunately, MPI includes several subroutines to accomplish the distribution of messages in an efficient way and they are independent of the machine's topology. Probably, the most important and constructive MPI-subroutines in this context are:

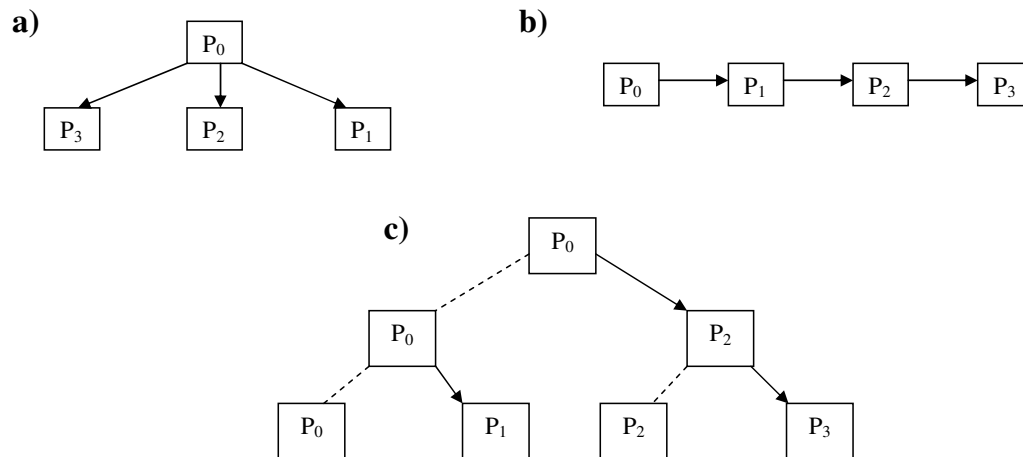


Figure 3.4: Various ways of distributing a message on 4 nodes. a) Classic way of distributing, where the master node P_0 sends the message to all processors in the communicator. b) chain scheme, in which each processor sends the message to its right neighbouring processor. c) Tree-Structured, the most effective way to distribute a message among several processors. In this example, P_0 sends the message only to P_2 and P_1 . P_2 in its turn sends the message to P_3 .

- **MPI_Bcast.** This function broadcasts a message from one processor to all other processors located in the same communicator (i.e. common world communicator), including itself.
- **MPI_Gather.** When calling the MPI-Gather, each processor sends the contents of its send buffer to a single processor (i.e. the root processor). The root processor receives these messages and stores them in its receive buffer according to the ID order of the senders.
- **MPI_Scatter.** The MPI-Scatter performs the reverse operation of the MPI-Gather function described above. The root processor divides its send buffer into n equal segments and sends segment 1 to processor of ID 1, segment 2 to processor of ID 2 and segment n to processor of ID n .
- **MPI_Alltoall.** Each processor in the communicator performs a scatter operation, sending a separate message to all processors in the group ordered by the processor's ID.

These MPI-subroutines are part of the collective communication routines. In Point-to-Point communication we can distinguish between two main categories: Blocking and non-blocking message passing. The latter is used if overlapping between communication and computation is needed. This means that the processor that sends a message will not wait until the send operation is completed, it can enter the computation phase and check later if the send operation is completed. Figure 3.5 shows

the difference between these two categories. For the parallel schemes developed in this thesis, blocking message passing routines are intensively used. The term blocking refers to the computation process, because between send or receive messages, the computation is blocked on both processors.

The most commonly used MPI blocking message passing routines are:

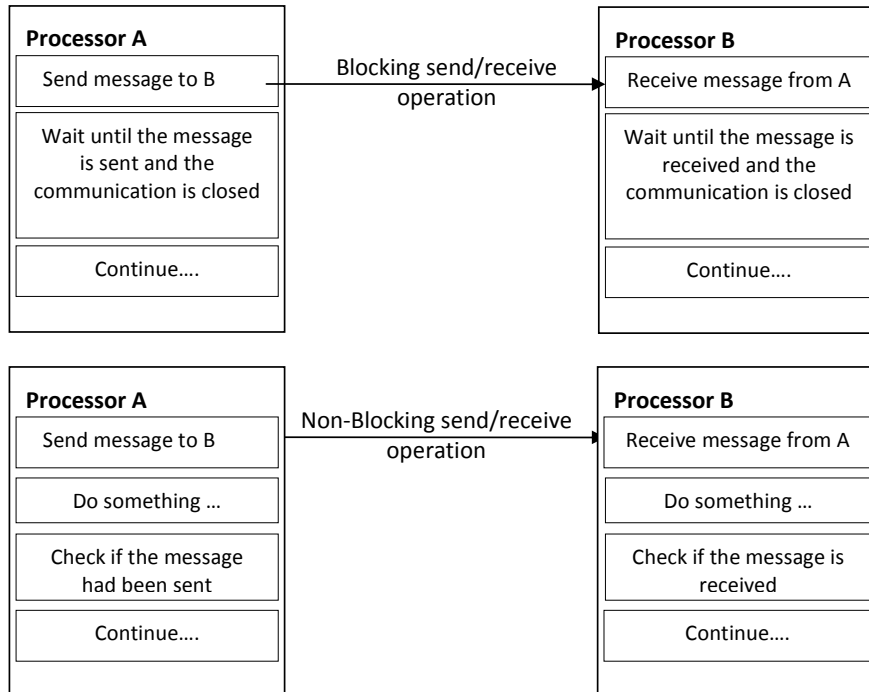


Figure 3.5: Differences between blocking and non-blocking send and receive operations via MPI Point-to-Point. In blocking send/receive operations, processor A will wait until processor B has completely received the message. Afterwards, both processors can enter the computation stage. In non-blocking send/receive operation, both processors will not wait until the send/receive operations are completed, they enter the computation stage and check later if the send/receive operations have been completed.

- **MPI_Send.** Basic blocking send operation. This routine returns only after the buffer of the sending processor is free for reuse.
- **MPI_Recv.** Receive a message and block until the requested data is available in the buffer of the receiving processor.
- **MPI_Bsend.** Buffered blocking send. This routine allocates the required amount of buffer space for the data of the sending processor. This routine insulates against problems associated with insufficient system buffer space. Before using

`MPI_Bsend` the memory must be allocated using `MPI_Buffer_attach`. After accomplishing the `MPI_Bsend` operation, the memory is released using `MPI_Buffer_detach` and then can be used by other operations.

Other important MPI routines that are part of many MPI programs are the so called MPI environment management routines. These routines are responsible for initializing the common world communicator (`MPI_Buffer_Init`), to determine the number of processors associated with it (`MPI_Buffer_size`), and to determine the IDs of the calling processors (`MPI_Buffer_rank`). Initially, each processor will be assigned a unique integer ID between 0 and number of processors-1.

In the following exemplary FORTRAN program, a matrix-matrix multiplication is computed in parallel to demonstrate the functionality of MPI routines. The main task is to compute the cross product $\mathbf{A} * \mathbf{A}^T = \mathbf{C}$, where \mathbf{A} and \mathbf{C} are two real matrices with the dimension $(n \times n)$ and T denotes the transpose of \mathbf{A} . Explaining and verifying the computation speed of the parallelized cross product using this example is very important, since the cross product ($\mathbf{J} * \mathbf{J}^T$ or $\mathbf{J} * \mathbf{J}^T$) consumes most of the time in MT inversion.

To simplify the explanation of this code, we will assume that n processors are available, processor with ID 0 will be the master and the rest will be considered as workers (Fig. 3.6 and Fig. 3.7).

To test the efficiency of the cross product MPI-Fortran code, its total run time (communication and computation) is measured and compared with the total run time of the serial version of the code (using only one processor). To evaluate the speed of both versions of the code, we introduce the relative speed-up factor (`Serial_Time/Parallel_time`) using different numbers of matrix elements ($n \times n$) with a fixed number of processors (100 processors) for the parallelized version. The red curve in Fig 3.8 shows the results. As can be expected, for matrices with less than 10^6 elements ($n < 1000$) the serial version is faster than the parallel code due to the time wasted by the communications. However, the speed-up factor increases dramatically for matrices with more than 10^6 elements ($n > 1000$). Obviously, for large matrices, the determining factor is the computation time and not the communication time.

The previous code was written using collective communication routines (`MPI_Bcast`, `MPI_Gather` and `MPI_Scatter`) to distribute data from the master processor to other processors in the communicator (`MPI_COMM_WORLD`), and to collect results from workers on the master. The alternative option to establish communication between processors is the Point-to-Point communication routines. To compare the efficiency of these two types of communication routines, the previous code was rewritten by replacing all collective communication routines with Point-to-Point communication routines:

- `MPI_Bcast(A(1,1),matrix_size,...)` routines in the master task and in the worker task were replaced by `MPI_SEND(A(1,1),matrix_size,...)` and

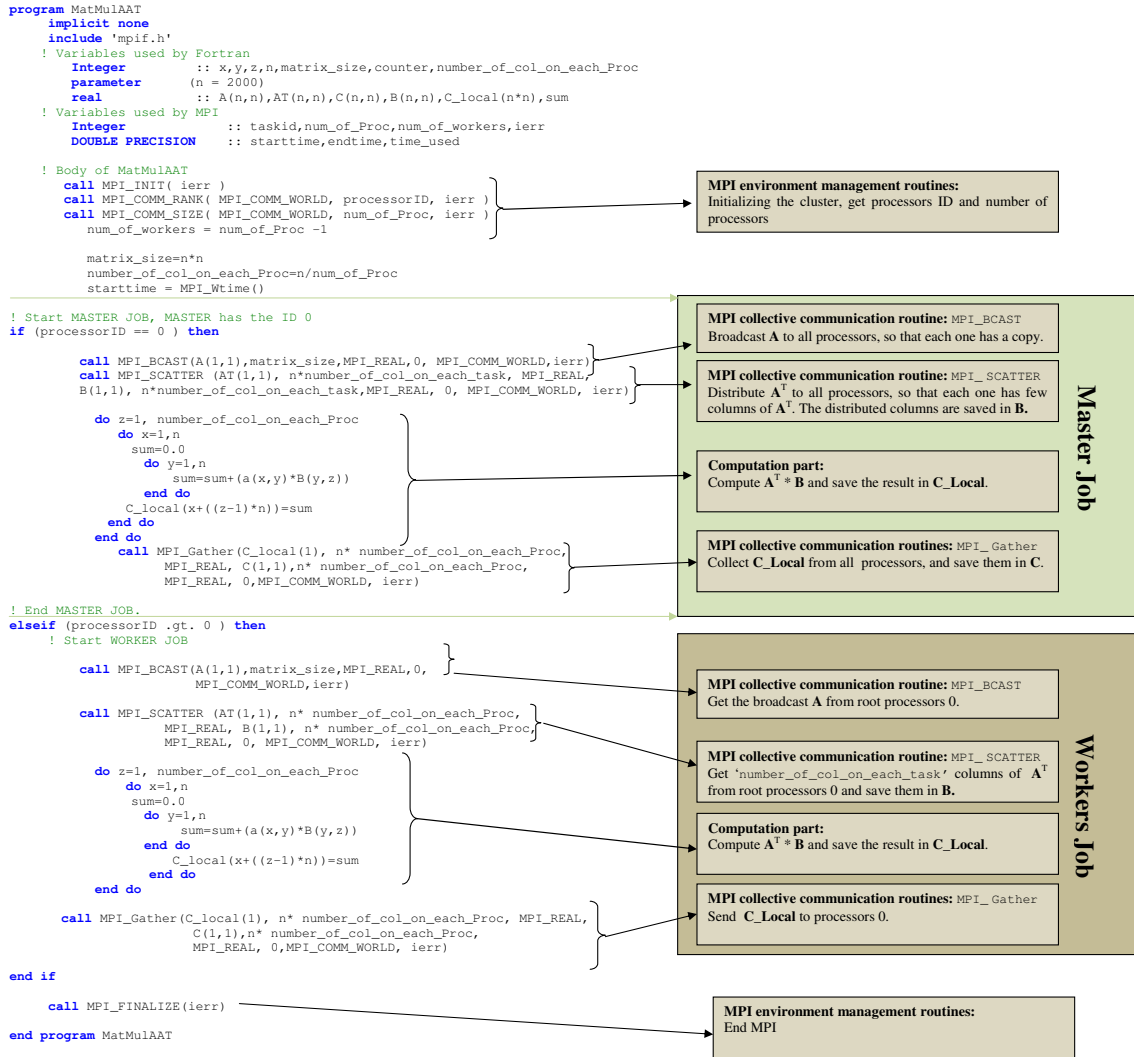


Figure 3.6: Parallelized FORTRAN code to calculate the cross product $\mathbf{A} * \mathbf{A}^T$

`MPI_RECV(A(1,1),matrix_size,...)`, respectively.

- Similarly, `MPI_SCATTER(AT(1,1),n*number_of_col_on_each_task,...)` and `MPI_Gather(C_local(1),n*number_of_col_on_each_task,...)` routines in the master task and in the worker task were replaced by: `MPI_RECV(C(1,which_col),n*number_of_col_on_each_task,...)` and `MPI_SEND(C_local(1),n*number_of_col_on_each_task,...)`, respectively.

The modified version of the code using Point-to-Point communication routines was executed using the same parameters (fixed number of processors and varying matrix dimension from 100×100 to 12800×12800). Figure 3.8 (blue curve) shows the results of this test. Point-to-Point communication behaves similar to the collective commu-

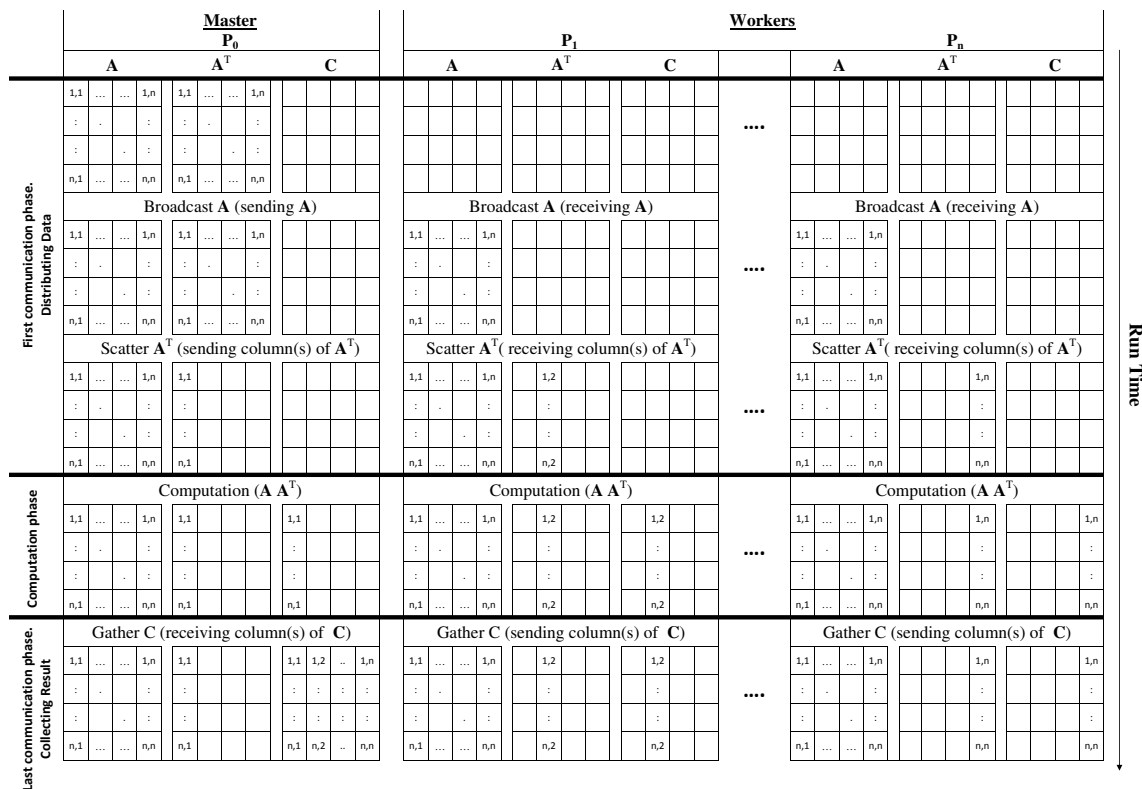


Figure 3.7: The communication and computation tasks used to carry out the multiplication $\mathbf{A} * \mathbf{A}^T$. In the first communication phase the master processor P_0 fills the corresponding locations of \mathbf{A} and \mathbf{A}^T on the memory of all workers after each call to one of the collective communication routines. Afterwards, each processor carries out the multiplication of \mathbf{A} with the distributed columns of \mathbf{A}^T . Once the computation is finished, each processor sends its local result to the master processor P_0 : Last communication phase.

nication when compared to the serial version of the code. For small matrices, most of the total time is consumed by communication, consequently, the serial version is faster than the parallel code. On the contrary, when using large matrices the communication time is short in comparison to the computation time, and therefore the parallelized version is faster than the serial code.

A comparison between collective and Point-to-Point communication routines (red and blue curves in Fig. 3.8) indicates another interesting result: For large matrices, Point-to-Point communication routines are faster in sending the data and receiving the results than collective operations.

The previous code example was written using the classical way to calculate the cross product multiplication (row by column). To optimise both computation and communication aspects of the code, certain properties of the cross product multiplication can be used. For example, multiplying \mathbf{A} with its transpose \mathbf{A}^T is the same as multiplying

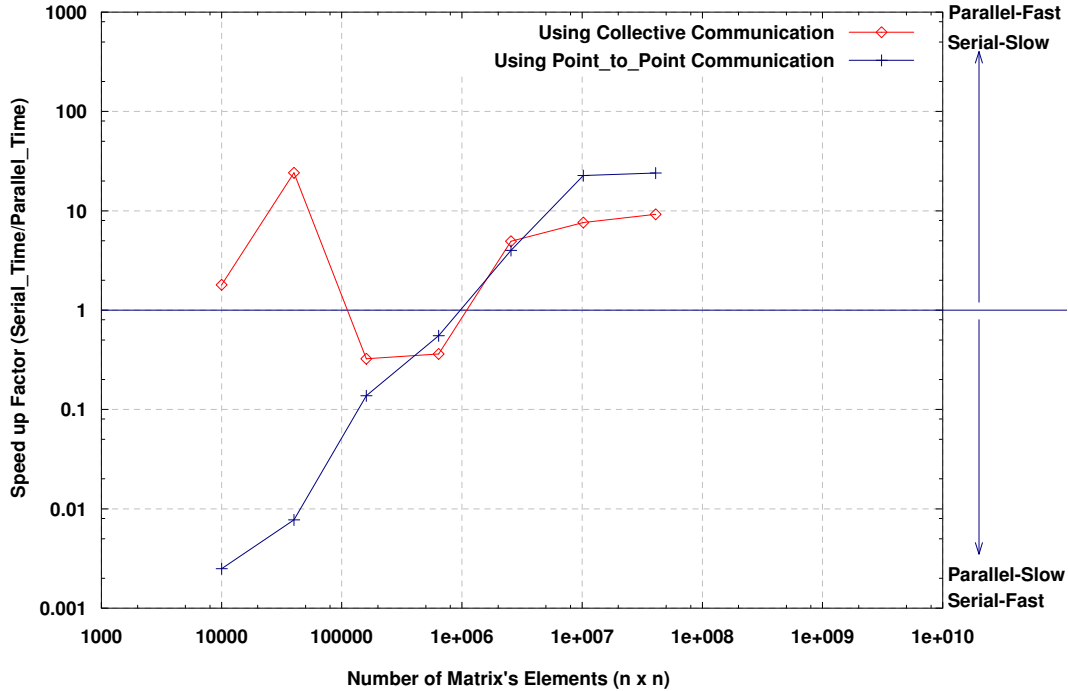


Figure 3.8: Comparison between serial and parallel cross product computation. A speed up factor of 1 means that serial computation is as fast as parallel execution. Red curve shows the result using collective communication, blue curve is for point_to_point communication. For large matrices ($n > 100$) the parallel computation is significantly faster than serial computation. Furthermore, a comparison between the collective and point_to_point communication reveals that the latter is faster in transferring the data between the processors especially for large matrices.

each row of \mathbf{A} with all rows of \mathbf{A} . It is therefore not necessary to broadcast \mathbf{A}^T , since \mathbf{A} was already broadcasted to all processors. Thereby, the time for communication is reduced as one `MPI_Bcast` (or one pair of `MPI_SEND` and `MPI_RECV`) routine can be removed from the code. Another optimization step can be achieved by taking into account that the cross product multiplication results in a symmetric matrix \mathbf{C} . Therefore, only the lower or upper triangular matrix is needed to construct the full matrix. This consideration is also valid for the serial algorithm. But for the parallel code, a suitable distribution of tasks among processors must be found. Otherwise, some processors do most of computations, while others spend most of the time waiting. This becomes obvious if we examine the following scheme to compute, e.g., the lower triangular matrix of \mathbf{C} (Table 3.1 and Fig. 3.9)

Table 3.1 and Fig. 3.9a indicate that P_1 will be busy the whole time to accomplish n row by row multiplications. At the same time, P_n does only one row by row multiplication. This simple example demonstrates very clearly that parallelization requires more numerical concepts. To make the algorithm efficient, all participating processors must have the same number of row \times row multiplications.

Processor ID	task	rang	# of rows multiplication
P_1	$row_1 * row_i$	$i = 1, \dots, n$	n
P_2	$row_2 * row_i$	$i = 2, \dots, n$	$n - 1$
\vdots	\vdots	\vdots	\vdots
$P_{n/2}$	$row_{n/2} * row_i$	$i = n/2, \dots, n$	$n/2$
\vdots	\vdots	\vdots	\vdots
P_n	$row_n * row_i$	$i = n, \dots, n$	1

Table 3.1: Distribution of row multiplication tasks among n processors to achieve the computation $\mathbf{A}\mathbf{A}^T$. P_1 multiplies the first row with all rows having indices greater than or equal to 1. Thus, P_1 must achieve n row \times row multiplications. On the other hand, processor P_n has only one row \times row multiplication.

Figure 3.9b shows the distribution of row \times row tasks among n processors after modification. This modification is based on the symmetrical properties of the final result (the matrix \mathbf{C}). In the modified task distribution scheme, each processor must carry out $n/2$ row \times row multiplications, where n is the number of rows in \mathbf{A} . To simplify the description of the modified version, let us assume that n row by row multiplications must be achieved by n processors to compute the lower triangular matrix of \mathbf{C} . Processors with IDs from 1 to $n/2$ (P_1 to $P_{n/2}$) compute $n/2$ row \times row computations to produce $n/2$ columns and $n/2$ rows (starting from the main diagonal) of \mathbf{C} . The missing elements in the lower triangular matrix of \mathbf{C} (to the left of $n/2$ column) are computed by processors $P_{(n/2)+1}$ to P_n because their original computation load is less than $n/2$. For example, the missing $(n/2)+1$ element in \mathbf{C} (dark green colour box in fig. 3.9b) is computed by processor $P_{(n/2)+1}$, after the processor has accomplished its original task to achieve $(n/2)-1$ row \times row computations.

3.3 Solving the MT forward modeling in parallel

The knowledge gathered on parallel computing using MPI will now be applied to solve the MT forward modelling problem in 2D and 3D.

The reason why parallel computing schemes must be applied for MT forward modelling, especially in 3D, is the large number of model parameters for which the electromagnetic (EM) solution must be approximated numerically using finite differences (FD) or finite elements (FE). The EM solution in 3D consists of the three components of the primary electric field \mathbf{E} (E_x , E_y and E_z) and three components of the secondary magnetic field \mathbf{B} (B_x , B_y and B_z). Thus, the number of elements of the approximated EM solution can easily exceed the limits of the computer memory. Even if the required memory is available, the time required to solve the problem in a serial manner

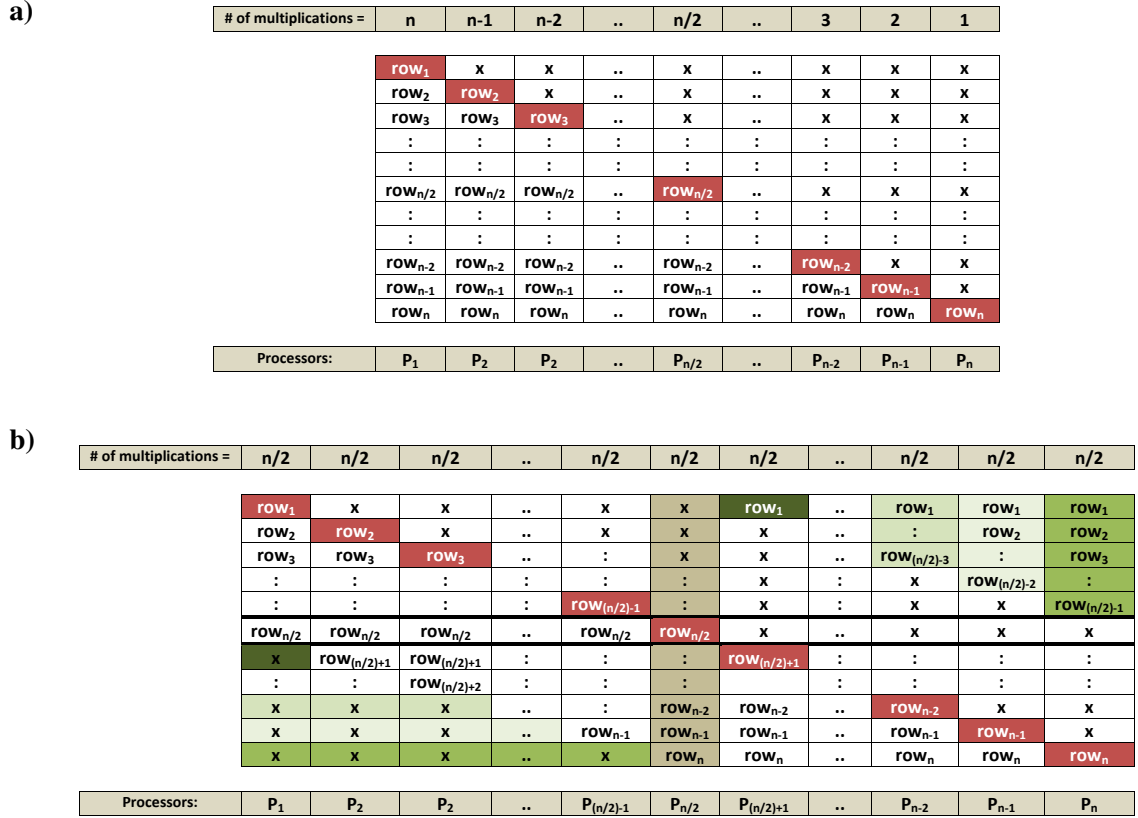


Figure 3.9: a) The lower triangular matrix of \mathbf{C} computed with n processors. The multiplication tasks are not evenly shared between the processors. b) The lower triangular matrix of \mathbf{C} , after modifying the distribution of the multiplication tasks. In this case, each processor has an equal number ($n/2$) of row \times row multiplications. For example, P_1 that has originally n row \times row multiplications to construct the first column of \mathbf{C} , must compute only $(n/2)$ row \times row. The missing elements in the first column are computed by another processors. For example, the element in the dark green box in the first column will be computed by processor $P_{(n/2)+1}$ instead of processor P_1 .

is usually too long. Therefore, a parallel computation of the MT forward solution must address both the time and memory problems.

The application of parallel computing in electromagnetic methods to approximate numerically Maxwell's equations (2.1 to 2.6) in time or in frequency domain is relatively new. However, solving sets of linear equations (similar to 2.49) is a typical problem for FD applications and one of the main motivations for developing new strategies and methods by the parallel computing community.

Earlier contributions in the context of solving the EM forward modelling problem in parallel, include the work of Alumbaugh *et al.* (1996). In this work, the authors present a method for modelling the wideband, frequency domain electromagnetic responses of a 3D earth model using a dipole source operating in frequency range from

100Hz to 30MHz. The main issue in that paper is the application of “Multiple Instruction Multiple Data (MIMD)” computers to approximate the modified form of the vector Helmholtz equation on a staggered grid using the FD method. The forward modelling parallel computing scheme is based on distributing the model parameters to all processors, so that each one works on a small portion of the model to construct the coefficients matrix \mathbf{A} in eq. 2.49 and to solve the equation system for the scattered electric field. However, a massive amount of communication is required in this scheme to exchange information between processors.

An efficient way to parallelize the MT forward modelling problem (and any other computation process) is to decompose the problem into its fundamental computational steps and then examine the dependencies between these steps. The MT forward modelling problem can be decomposed into:

1. Setting the input parameters, including model parameters, station locations and sets of periods for which the EM solutions must be approximated.
2. Formulating a discrete representation of the gradient (∇), divergence ($\nabla \cdot$) and curl ($\nabla \times$) operators on a staggered grid using finite differences.
3. Constructing the coefficients matrix \mathbf{A} from the previous differential operators, including the time dependency term ($e^{-i\omega t}$) on the main diagonal of \mathbf{A} and the model parameter distribution (see section 2.5 for more details).
4. Creating the right hand side (\mathbf{b}) of equation 2.49 which comprises the boundary conditions and sources. Since MT uses natural EM fields, the source term is the normal fields generated by 1D layered earth modes (2D case) or 2D earth models (3D case).
5. Solving the equation system 2.49 for \mathbf{x} and a given set of periods. In the 2D case, \mathbf{x} represents \mathbf{E}_x or \mathbf{H}_x for the TE and TM modes, respectively. In the 3D case, the equation system must be solved for two different polarizations or modes (E_x-H_y and E_y-H_x) (see sections 2.4 and 2.5).
6. Interpolating the computed EM solution to the station locations to calculate the model responses at the given locations.

In general, any forward modelling scheme contains these steps. The main difference between the 2D and 3D cases is in the construction of the coefficients matrix \mathbf{A} (step 3), which has different dimensions and structures in both cases.

To parallelize the steps above, one has to examine their dependency on each other and to define the required tasks which can be computed on each processor independently. Steps 1 to 4 do not require much memory and computation time. Therefore, they can easily be accomplished on a single processor, say the master processor. The main computational effort takes place in step 5, in which the equation system is solved for

a number of periods and modes. Since the solution of 2.49 for one period does not depend on the solution of the other periods, eq. 2.49 can be solved independently for one particular period. Furthermore, the solution of different modes can also be done independently, since the solution of each mode does not depend on the other mode. In summary, the parallelization scheme to solve the forward modeling problem in the 2D and 3D cases can be addressed as follows: The master processor reads the input parameters and creates \mathbf{A} and \mathbf{b} , sends \mathbf{A} , \mathbf{b} and the index of one period and one mode to individual worker processors. Each worker processor receives the required parameters to solve the equation system 2.49. Once the workers finish their computation task, they send a ready signal to the master. Afterwards, the master starts receiving the model responses from each worker.

In the 3D case, computing the model responses at a given location for one particular period requires the solution from both modes. Consequently, a communication between the processors takes place to exchange solutions from different modes. Let us consider the following example: Suppose that we want to solve the 3D MT forward modelling problem using the model parameters \mathbf{m} , two periods (Per_1, Per_2) and ten stations ($Stn_i, i = 1, \dots, 10$) for which the model responses must be calculated. Assume that there are five processors available (P_0, \dots, P_4 in Fig. 3.10). The

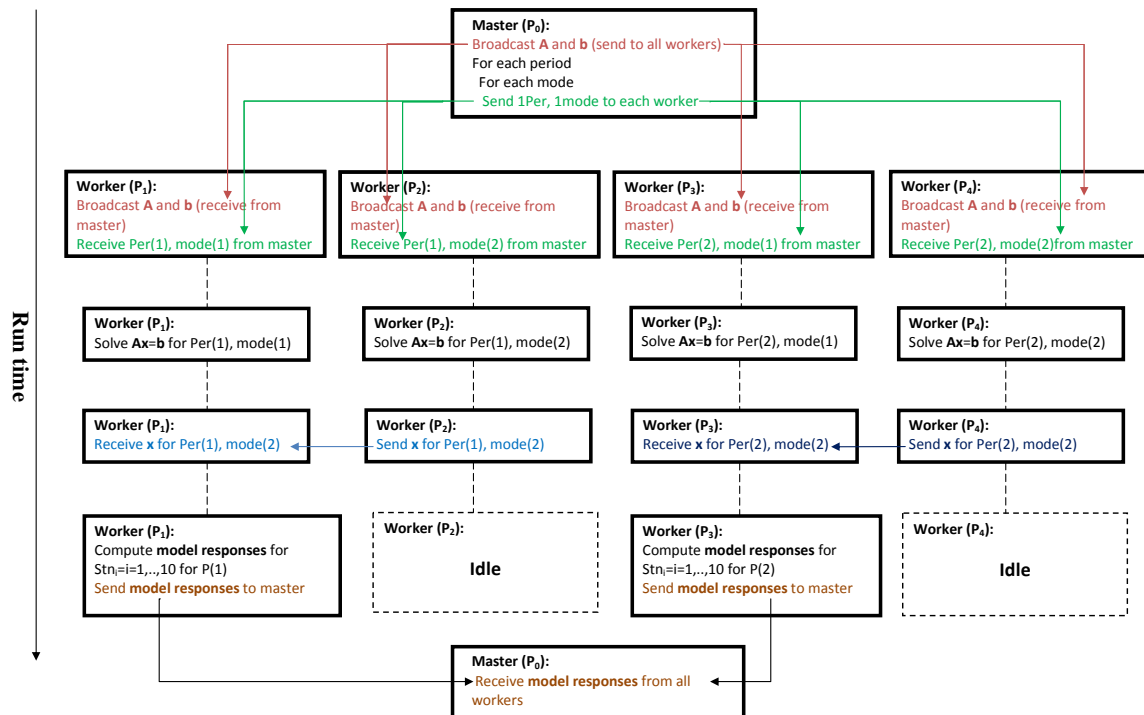


Figure 3.10: Parallelization scheme to solve the 3D MT forward modelling problem. The example demonstrates the communication as well as the computation steps to compute the model responses using five processors (P_0, \dots, P_4).

presented scheme to solve the forward modelling problem in parallel has the following advantages:

- The time required to solve the forward problem is reduced to $t/NumProc$, where t is the time required to solve the problem with one processor (serial code) and $NumProc$ is the number of processors. Ideally, $NumProc$ is equal to $NPer * 2$, where $NPer$ is number of periods and 2 stands for number of modes.
- The memory space required for the EM solution (the result after step 5) is reduced by the factor $NPer$, because the EM solution is distributed between all processors, and only the solutions of the two modes need to be stored on the memory of each processor².
- The communication between processors is kept at a minimum. Only one pair of processors communicates with each other to send and receive data. Additionally, only half of the workers will communicate with the master to send the model responses as messages, which is small in comparison to the entire EM solution message that consists of the electric field components (E_x, E_y, E_z) at the edges of each model parameter and for all periods.

3.4 An efficient scheme to invert MT data in 2D/3D in parallel

Inverting MT data in 3D in parallel is a relatively new issue in the electromagnetic community. Earlier work in this context is published by Newman & Alumbaugh (1997), where they used massively parallel (MP) computers to solve the 3D inverse problem for controlled source electromagnetic (CSEM) method. The applied parallel scheme for the 3D inversion of CSEM data based on the same principle used to solve the 3D forward modelling problem published by Alumbaugh *et al.* (1996). However, Newman & Alumbaugh (1997) used the conjugate gradients method (CG) to solve iteratively the normal equation results from the inverse problem. Later, Newman & Alumbaugh (2000) used the same concept to invert 3D MT data in parallel.

A recent study published by Siripunvaraporn & Egbert (2009) extends the serial version of the 3D MT inversion code (WSINV3DMT) to run in parallel. The used inversion scheme in WSINV3DMT is the occam inversion, however in data space (see sec. 2.6.5; Siripunvaraporn *et al.* (2005)). The parallel scheme proposed by Siripunvaraporn & Egbert (2009) is more appropriate for clusters with few processors. This parallel scheme is based on distributing the forward modelling as well the sensitivity matrix computations on several processors. However, each processor solves the

²Assuming that the number of processors is equal to $NPer * 2$.

forward modelling problem and computes the sensitivity matrix for a sub set of periods. This implies that constructing the cross product matrix requires a massive communication between processors to exchange the subsets of the sensitivity matrix (Siripunvaraporn & Egbert, 2009). The constructed normal equation system is then solved for a model update using Cholesky decomposition if the number of data is small. Otherwise, a parallel iterative solver (i.e. PCG) is used to solve the normal equation for a model update (Siripunvaraporn & Egbert, 2009).

Most recently, Commer & Newman (2009) applied the method of nonlinear conjugate gradient (NLCG) to solve the joint 3D inverse problem for CSEM and MT in parallel. The proposed parallel scheme based on the work published by Alumbaugh *et al.* (1996). However, several strategies on different levels of parallelization are combined to optimise the solution of the 3D joint inversion. For the forward problem of CSEM and MT data, a fast iterative QMR solver was adopted by Commer & Newman (2009). Furthermore, Commer & Newman (2009) applied a grid-optimization strategy to limit the FD mesh.

The parallelization schemes presented in this section are based on the mathematical formulation of the inversion process discussed in section 2.6.

Following the same strategy as in solving the forward modelling problem in parallel by decomposing the problem in several steps, the inversion process can be decomposed as follows:

1. Setting the inversion parameters (e.g. number of iterations, desired R.M.S., starting value of the regularization parameter,..., etc).
2. Solving the forward modelling problem to compute the residuum between the measured and predicted data (eq. 2.55).
3. Compute the sensitivity matrix \mathbf{J} if it is required or compute the multiplications $\mathbf{J}^T \mathbf{d}$ and $\mathbf{J} \mathbf{m}$ (see section 2.6 for more details).
4. Solving the normal equation in the model space (i.e. eq. 2.71) or in the data space (eq. 2.105).
5. Repeat steps 2 to 4 until a stopping criterion is reached (e.g. maximum number of iterations or the residuum is less than the given value).

Some of the steps discussed above depend on the used inversion scheme. Many inversion schemes (e.g. Gauss-Newton, Marquard-Levenberg and Occam inversion) require an explicit expression of the sensitivity matrix \mathbf{J} at each iteration to solve the normal equation for a model (\mathbf{m}) or a model update ($\delta \mathbf{m}$) (see section 2.6). Furthermore, inversion schemes based on an explicit use of \mathbf{J} also require the cross product multi-

plications $\mathbf{J}\mathbf{J}^T$ or $\mathbf{J}^T\mathbf{J}$ ³. Computing the sensitivity matrix and computation of the cross product are the most time and memory space consuming steps in the inversion schemes that require an explicit expression of the sensitivity matrix \mathbf{J} . Other widespread inversion schemes, especially in 3D, are the conjugate gradient method (CG) to solve the linearized normal equation and the non linear conjugate gradient method (NLCG) to solve directly the non linear inverse problem. In both methods the explicit use of \mathbf{J} is avoided by forming the matrix-vector products ($\mathbf{J}^T\mathbf{d}$ and $\mathbf{J}\mathbf{m}$) which help to reduce the required memory space (see section 2.6). However, computing these two particular matrix-vector products is very time consuming, when calculated on a single processor.

To keep the developed MPI-Module in this thesis as generic and multipurpose as possible, the next two subsections describe solutions for parallelizing the computation of the sensitivity matrix and the cross product.

3.4.1 Computing the sensitivity matrix in parallel

The calculation of the sensitivity matrix in parallel is complicated, as both computation time and memory space must be optimized. Furthermore, various mathematical operations must be carried out using the entries of the distributed sensitivity matrix making the procedure of dealing with this matrix more complicated.

To clarify the concept of parallelizing the computation of the sensitivity matrix, let us have a look on the structure of this matrix: By definition, the sensitivity matrix describes how sensitive predicted data points⁴ are with respect to small changes of the model parameters (see section 2.6.2). Therefore, the number of columns of \mathbf{J} is equal to the total number of model parameters, say M , and the number of rows is equal to the total number of predicted data points, say N (Fig 3.11).

Figure 3.11 demonstrates the structure of the sensitivity matrix by using two periods (Per_1 and Per_2), three stations (Stn_1 , Stn_2 and Stn_3) and eight model responses representing the real and imaginary parts of the impedance tensor at each station⁵. Hence, each block of eight rows in \mathbf{J} relates to one station and one period. This structure of \mathbf{J} allows dealing with the matrix block-wise. Each block contains a number of rows equal to the number of model responses to be inverted, multiplied by two to allow for real and imaginary parts (or apparent resistivity and phase).

Section 2.6.2 addresses the mathematical formulation to compute the entries of the sensitivity and describes several characteristics of \mathbf{J} . Similar to solving the forward modelling problem in parallel, the computation of \mathbf{J} for a particular period, will not

³Depending if the inversion problem is solved in model or data space, respectively

⁴The predicted or measured data can be presented as real and imaginary parts of the impedance tensor or as apparent resistivity and phase.

⁵Other model responses could be also included, e.g. the vertical and horizontal magnetic transfer functions. In this case the number of rows in each block expands according to the number of required model responses

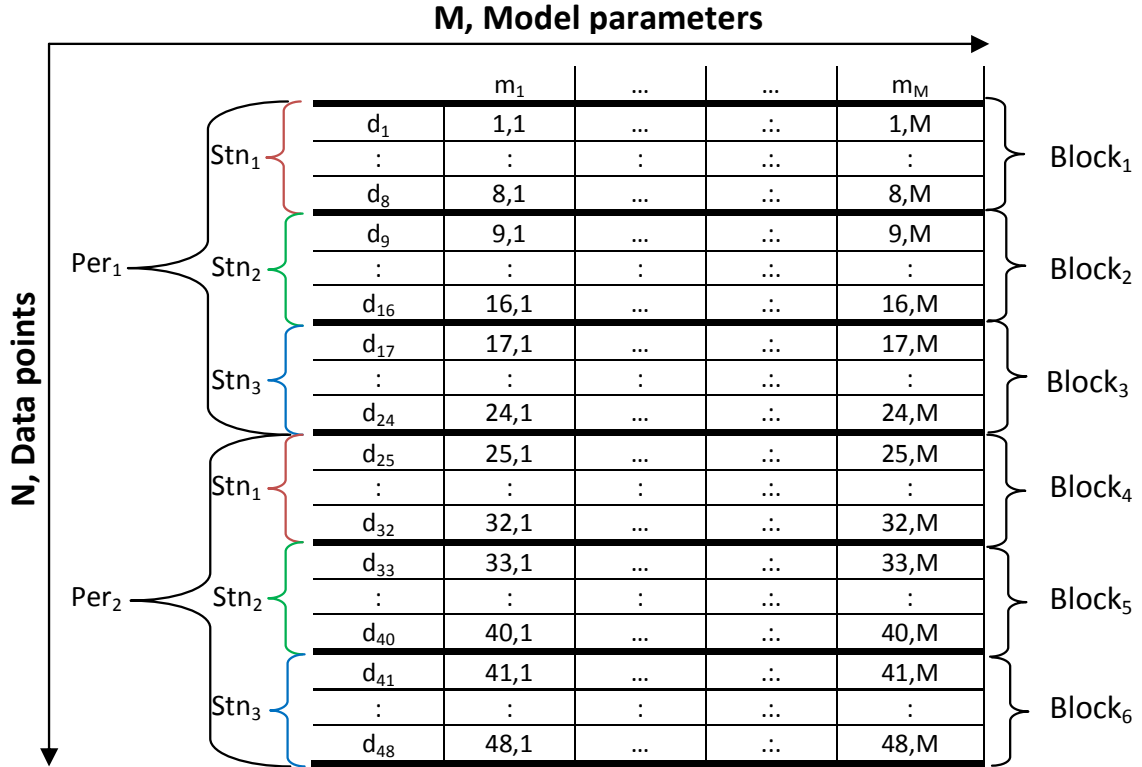


Figure 3.11: The structure of the sensitivity matrix (\mathbf{J}) when using M model parameters and 48 data points: 2 periods \times 3 stations \times 8 real and imaginary parts of the impedance tensor. The matrix can be presented block-wise. Each block contains the model responses associated with one period and one station. In this example, \mathbf{J} contains 6 blocks (2 periods \times 3 stations).

depend on the computation of other periods. Therefore, one can parallelize over periods. Furthermore, the computation of several rows (blocks in fig. 3.11) of \mathbf{J} , which correspond to one station is independent from the computation of other rows (corresponding to other stations). Hence, one can also parallelize over stations. This means that each processor will be responsible for the computation of one block of \mathbf{J} (Fig. 3.12).

The computation of \mathbf{J} requires the existence of the forward solution, in particular the electric and magnetic fields (two modes in 3D) on all nodes or edges of the model resulting from the forward modelling step (see section 2.6.2). That is, if the complete EM solution for one period is kept in the memory of each worker, then each worker can compute the corresponding block of \mathbf{J} for one particular period and any station. Consequently, if we implement the forward modelling steps before computing \mathbf{J} , which is normally the case, then we can implement the parallelization scheme as follows: The master sends the indices of one period and one mode to each worker. Each worker computes the EM solution for one period and one mode. Communication between

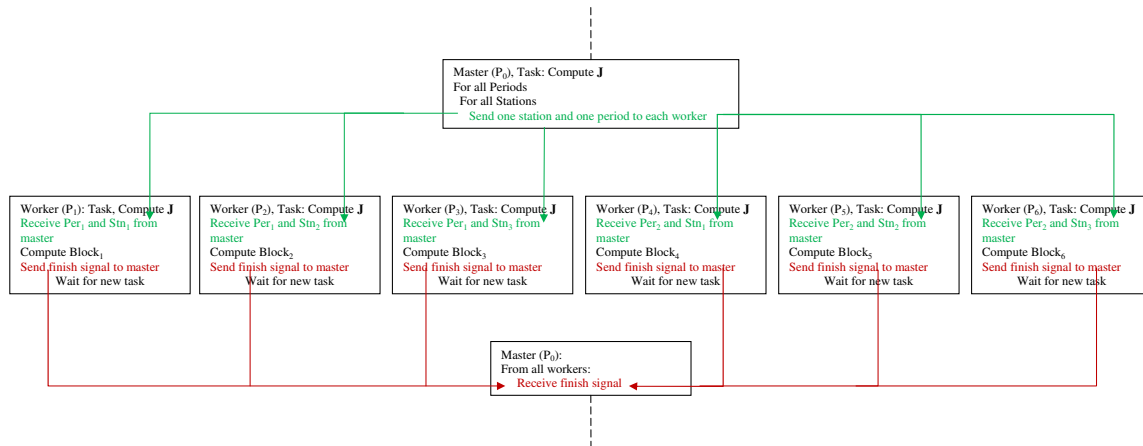


Figure 3.12: Task distribution among six processors to compute the blocks of the sensitivity matrix \mathbf{J} presented in Fig. 3.11. The master processor (P_0), as well as the worker processors (P_1 - P_6), have the task to compute \mathbf{J} . P_0 establishes a communication with each worker to send the indices of one period and one station. Each processor is responsible for the computation of one block of \mathbf{J} . Once the computation is finished, each worker sends a finish signal to the master.

two workers working on the same period takes place to exchange the EM solutions for the two different modes, so that each worker keeps a copy of the EM forward solution in its memory. The next task is to compute \mathbf{J} . Since the master knows which worker has the EM solution for a particular period, it sends that period index to the worker who is hosting the respective EM solution. Additionally, the master sends one station index to each of the workers; it can be any station index (Fig. 3.13).

Figure 3.13 illustrates the communication between the master and the workers for solving the forward modelling problem and subsequently computing the sensitivity matrix (\mathbf{J}).

3.4.2 Cross product $\mathbf{J}\mathbf{J}^T$ and $\mathbf{J}^T\mathbf{J}$ in parallel

Inversion schemes that make explicit use of Jacobi matrix (\mathbf{J}) require performing several mathematical operations with that matrix. The most important and time consuming mathematical operation is the cross product multiplication.

If the inverse problem is solved in the data space, the cross product $\mathbf{J}\mathbf{J}^T$ is computed, whereas in model space the cross product $\mathbf{J}^T\mathbf{J}$ is calculated. Carrying out any of these cross product multiplications in parallel is not a trivial task, since \mathbf{J} is distributed block- wise among all processors (Figs. 3.12 and 3.13). In both cases of cross product multiplications we can make use of the sample code for general matrix multiplication presented in Fig. 3.9.

Before addressing the cross product in detail, it is worthwhile to notice that in the computation stage of \mathbf{J} , each processor saves the index of the received period and sta-

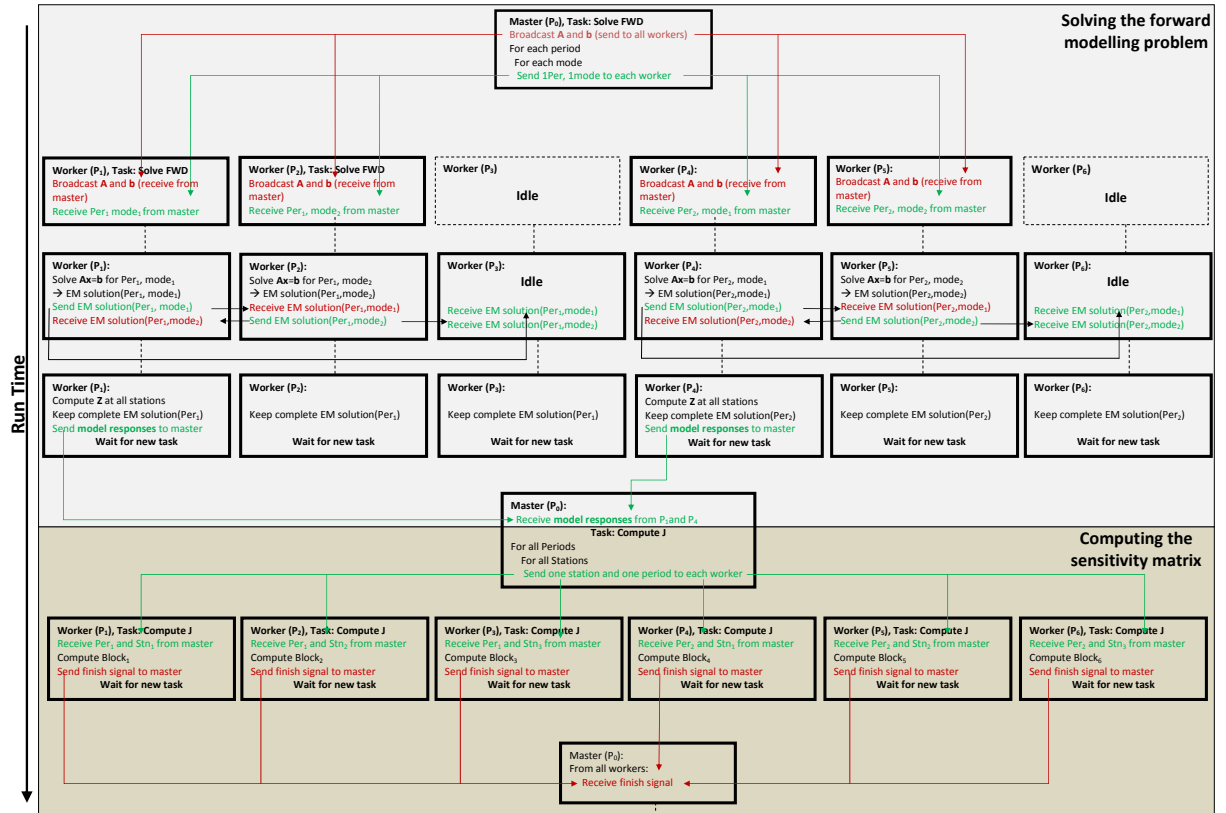


Figure 3.13: Task distribution among six processors to solve the forward modelling problem and subsequently compute the corresponding blocks of the sensitivity matrix (\mathbf{J}). The first task in this parallel scheme is to solve the forward modelling problem, which is now a part of the task “Compute_J”. After solving the forward modelling problem, each processor keeps the EM solution for one particular period in its memory. This can be done by exchanging the EM solutions for the two different modes between two processors (e.g. between P_1 and P_2) or by receiving the solutions for the two different modes from two different processors (e.g. P_3). \mathbf{J} is computed in parallel on all six processors: The master sends the index of one particular period to the worker who is hosting the respective EM solution and any station index. For example, the blocks corresponding to the first period are computed by processors P_1 - P_3 .

tion, as these indices are required later to compute the block index. Once the parallel computation of \mathbf{J} is accomplished, each processor sends the block index to the master. The master receives these indices and saves them together with the processor ID. The master broadcasts this information to all workers, so that each worker is informed about the location of the blocks.

Several strategies can be adopted to achieve the cross product multiplication in parallel. However, which one to apply is a question of performance in terms of computation and communication. Therefore, several strategies will be discussed and their performance is examined in terms of computation and communication requirements.

Let us consider the following example: Suppose that we have two periods, seven stations, M model parameters and eight processors. Computing the sensitivity matrix (\mathbf{J}) in parallel requires distributing the computation of the matrix block-wise between all processors. Since we have 14 blocks (2 periods \times 7 stations) but only 7 processors as workers, each processor must compute and save two blocks of \mathbf{J} (i.e. P_1 computes and saves $block_1$ and $block_8$). Dealing with \mathbf{J} block-wise means that the multiplications occur between the blocks rather than between rows. To keep track which rows have been multiplied with each other on which processor, each processor saves the index of the multiplied rows in two vectors and the multiplication values in a third vector. The master receives these vectors from each processor to sort and save the final result of the cross product multiplication $\mathbf{J}\mathbf{J}^T$.

Let us now consider how the cross product $\mathbf{J}\mathbf{J}^T$ is computed in parallel. The simplest method to compute $\mathbf{J}\mathbf{J}^T$ on one processor requires two loops. The first loop runs over all blocks ($B_i, i = 1, \dots, Nblocks$) and the second loop is also over all blocks but starting from i ⁶ ($B_j, j = i, \dots, Nblocks$). In the first loop the master requests from the processor who owns $block_i$ to return the block. In the second loop, the master asks for $block_j$ to compute the multiplications $block_i \times block_j$. Obviously, this scheme is very inefficient, because only the communication is parallelized while the computation is in serial.

To move one step further in the direction of parallelizing the cross product multiplication of the distributed \mathbf{J} , each processor contributes to the multiplication. Each processor multiplies its blocks with all other blocks having indices equal to or smaller than its own blocks. This, however, requires communication between the processors to exchange their blocks. For instance, processor P_1 which owns the blocks (B_1 and B_8) must first send B_1 to all other processor. Afterwards, it can carry out its multiplications ($B_1 \times B_1$ and $B_8 \times B_1$). The next step is to multiply B_2 with all other blocks. Therefore, P_2 which owns B_2 starts sending B_2 to all other processors, starting with P_1 . Once P_2 has finished sending B_2 , it can start with the computation of $B_2 \times B_2$ and $B_9 \times B_2$, etc. (see fig. 3.14). Many computations can be skipped (the yellow coloured multiplications in fig. 3.14) as they have been carried out already

⁶Since each block will be multiplied with all blocks having indices greater or equal its index.

by another processor (i.e. $B_1 \times B_2$ is skipped on P_1 , since P_2 has already calculated $B_2 \times B_1$). By examining the communication and computation performance of this method we can deduce that the computation part is efficient, as it is carried out on several processors in parallel. However, the communication part is not optimal, since one processor must send its block to all other processors starting from the first processor P_1 . As a consequence, many processors are idle, waiting for communications to finish before they can enter the computation stage (3.14). All processors must enter an idle state until the busy processor has accomplished its communication and subsequently its computation tasks. For example, P_2 tries to send B_2 to P_1 (the first processor in the communicator), while P_1 is still busy distributing B_1 and it cannot enter the computation stage. For that reason all other processors must wait until P_1 is ready with its computational tasks before they can receive B_2 from P_2 (Fig. 3.14). Obviously, the communication aspects of this scheme require more attention.

As a first step, we can remove the restriction that the send loop on each processor

P_1 owns Blocks: B_1, B_8	P_2 owns Blocks: B_2, B_9	P_3 owns Blocks: B_3, B_{10}	P_4 owns Blocks: B_4, B_{11}	P_5 owns Blocks: B_5, B_{12}	P_6 owns Blocks: B_6, B_{13}	P_7 owns Blocks: B_7, B_{14}		
Send B_1 to P_2	Recv. B_1 from P_1		Idle	Idle	Idle	Idle	t_1	
Send B_1 to P_3	Compute $B_2 * B_1$						t_2	
Send B_1 to P_4	$B_9 * B_1$	Recv. B_1 from P_1	Recv. B_1 from P_1				t_3	
Send B_1 to P_5	Idle	Compute $B_3 * B_1$	Compute $B_4 * B_1$	Recv. B_1 from P_1			t_4	
Send B_1 to P_6		$B_{10} * B_1$	Compute $B_5 * B_1$	$B_{11} * B_1$	Compute $B_5 * B_1$	Recv. B_1 from P_1		t_5
Send B_1 to P_7					$B_{12} * B_1$	Compute $B_6 * B_1$	Recv. B_1 from P_1	t_6
Compute $B_4 * B_1$			Idle	Idle	Idle	$B_{13} * B_1$	Compute $B_7 * B_1$	t_7
Compute $B_8 * B_1$							$B_{14} * B_1$	t_8
Recv. B_2 from P_2	Send B_2 to P_1						t_9	
Compute $B_8 * B_2$ skipped	Send B_2 to P_2	Recv. B_2 from P_2	Recv. B_2 from P_2				t_{10}	
$B_8 * B_2$	Send B_2 to P_3	Compute $B_3 * B_2$	Compute $B_4 * B_2$	Recv. B_2 from P_2			t_{11}	
Idle	Send B_2 to P_4	$B_{10} * B_2$	$B_{11} * B_2$	Compute $B_5 * B_2$	Recv. B_2 from P_2		t_{12}	
Recv. B_3 from P_3	Send B_2 to P_5	Send B_3 to P_1		$B_{12} * B_2$	Compute $B_6 * B_2$	Recv. B_2 from P_2	t_{13}	
Compute $B_8 * B_3$ skipped	Send B_2 to P_6	Idle	Idle	Idle	$B_{13} * B_2$	Compute $B_7 * B_2$	t_{14}	
$B_8 * B_3$	Compute $B_9 * B_2$					$B_{14} * B_2$	t_{15}	
Idle	Recv. B_3 from P_3	Send B_3 to P_2					t_{16}	
	Compute $B_8 * B_3$ skipped	Send B_3 to P_3	Recv. B_3 from P_3				t_{17}	
	$B_8 * B_3$	Send B_3 to P_4	Compute $B_4 * B_3$	Recv. B_3 from P_3	Compute $B_5 * B_3$	Idle	t_{18}	
		Send B_3 to P_5	$B_{10} * B_3$	$B_{11} * B_3$	Recv. B_3 from P_3	Recv. B_3 from P_3	Idle	t_{19}
		Send B_3 to P_6	Send B_3 to P_1	Send B_4 to P_1	Compute $B_5 * B_3$	Compute $B_6 * B_3$	Recv. B_3 from P_3	t_{20}
Recv. B_4 from P_4	Idle	Compute $B_3 * B_3$	Send B_4 to P_2	$B_{12} * B_3$	$B_{13} * B_3$	Compute $B_7 * B_3$	t_{21}	
Compute $B_8 * B_4$ skipped	Recv. B_4 from P_4	Compute $B_9 * B_3$	Idle	Idle	Idle	Compute $B_{14} * B_3$	t_{22}	
$B_8 * B_4$	Compute $B_9 * B_4$	$B_{10} * B_3$					t_{23}	
:		Recv. B_4 from P_4	Send B_4 to P_3				t_{24}	
:		Compute $B_8 * B_4$ skipped	Send B_4 to P_5	Recv. B_4 from P_4			t_{25}	
:		$B_{10} * B_4$	Send B_4 to P_6	Compute $B_5 * B_4$	Recv. B_4 from P_4		t_{26}	
:			Send B_4 to P_7	$B_{12} * B_4$	Compute $B_6 * B_4$	Recv. B_4 from P_4	t_{27}	
:			Compute $B_4 * B_4$		$B_{13} * B_4$	Recv. B_4 from P_4	t_{28}	
:			$B_{11} * B_4$			Compute $B_7 * B_4$	t_{29}	
:						$B_{14} * B_4$		
:								

Figure 3.14: A straight forward procedure to compute the multiplication $\mathbf{J}\mathbf{J}^T$ in parallel. In this example, each processor owns some blocks of \mathbf{J} that must be multiplied with all other blocks. Starting with B_1 to B_{14} , each processor must confirm that it owns the required block. If a processor owns the block, it enters in a loop, starting from P_1 over all processors excluding itself and sends the required block to all other processors. All other processors are in idle state until they have received the required block. Due to the symmetrical properties of the cross multiplication, the computations coloured with yellow can be skipped, since they have been computed already by other processors.

starts from the first processor. Instead, each processor who has to send a block sends the block to all processors ahead and then returns to address the processors behind.

For example, P_2 will not send B_2 to P_1 as in the previous example, but sends B_2 to P_3, \dots, P_7 before sending B_2 to P_1 (Fig. 3.15). In this case, the processors waste less time waiting for new blocks before entering the computation phase. This simple consideration can minimise the idle time of the processors considerably.

In the previous two examples we still have the problem that one processor must send

P_1 owns Blocks: B_1, B_8	P_2 owns Blocks: B_2, B_9	P_3 owns Blocks: B_3, B_{10}	P_4 Owns Blocks: B_4, B_{11}	P_5 owns Blocks: B_5, B_{12}	P_6 owns Blocks: B_6, B_{13}	P_7 owns Blocks: B_7, B_{14}	
Send B_1 to P_2	Recv. B_1 from P_1		Idle	Idle	Idle	Idle	t_1
Send B_1 to P_3	Compute $B_2 * B_1$	Recv. B_1 from P_1					t_2
Send B_1 to P_4	Idle	Compute $B_2 * B_1$	Recv. B_1 from P_1				t_3
Send B_1 to P_5		$B_{10} * B_1$	Compute $B_4 * B_1$	Recv. B_1 from P_1			t_4
Send B_1 to P_6	Send B_2 to P_3	Recv. B_1 from P_2		Compute $B_6 * B_1$			t_5
Send B_1 to P_7	Send B_2 to P_4	Compute $B_3 * B_2$	Recv. B_2 from P_2	$B_{12} * B_1$	Recv. B_1 from P_1	Recv. B_1 from P_1	t_6
Compute $B_8 * B_1$	Send B_2 to P_5	$B_{10} * B_2$	Compute $B_4 * B_2$	Recv. B_2 from P_2	Compute $B_{13} * B_1$	Compute $B_7 * B_1$	t_7
Idle	Send B_2 to P_6	Idle	$B_{11} * B_2$	Compute $B_5 * B_2$	Recv. B_2 from P_2	$B_{14} * B_1$	t_8
	Send B_2 to P_7	Send B_3 to P_4	Recv. B_3 from P_3	$B_{12} * B_2$	Compute $B_6 * B_2$	Recv. B_3 from P_2	t_9
Recv. B_3 from P_2	Send B_3 to P_1	Send B_3 to P_5	Compute $B_4 * B_3$	Recv. B_3 from P_3	$B_{13} * B_2$	Compute $B_7 * B_2$	t_{10}
Compute $B_8 * B_3$ skipped	Compute $B_9 * B_2$	Send B_3 to P_6	$B_{11} * B_3$	Compute $B_5 * B_3$	Recv. B_3 from P_3	$B_{14} * B_2$	t_{11}
$B_8 * B_2$	Send B_3 to P_7	Send B_3 to P_7	Idle	$B_{12} * B_3$	Compute $B_6 * B_3$	Recv. B_3 from P_3	t_{12}
Recv. B_3 from P_3	Idle	Send B_3 to P_1	Send B_4 to P_5	Recv. B_4 from P_4	$B_{13} * B_3$	Compute $B_7 * B_3$	t_{13}
Compute $B_8 * B_3$ skipped	Recv. B_4 from P_3	Send B_3 to P_2	Send B_4 to P_6	Compute $B_5 * B_4$	Recv. B_4 from P_4	$B_{14} * B_3$	t_{14}
$B_8 * B_3$	Compute $B_9 * B_3$ skipped	Compute $B_3 * B_3$	Send B_4 to P_7	$B_{12} * B_4$	Compute $B_6 * B_4$	Recv. B_4 from P_4	t_{15}
Recv. B_4 from P_4	$B_9 * B_3$	$B_{10} * B_3$	Send B_4 to P_1	Idle	$B_{13} * B_4$	Compute $B_7 * B_4$	t_{16}
Compute $B_8 * B_4$ skipped	Recv. B_4 from P_4	Idle	Send B_4 to P_2	Send B_5 to P_6	Recv. B_5 from P_5	$B_{14} * B_4$	t_{17}
$B_8 * B_4$	Compute $B_9 * B_4$ skipped	Recv. B_5 from P_4	Send B_4 to P_3	Send B_5 to P_7	Compute $B_6 * B_5$	Recv. B_5 from P_5	t_{18}
Recv. B_5 from P_5	$B_9 * B_4$	Compute $B_3 * B_4$ skipped	Compute $B_4 * B_4$	Send B_5 to P_1	$B_{12} * B_5$	Compute $B_7 * B_5$	t_{19}
Compute $B_8 * B_5$ skipped	Recv. B_5 from P_5	$B_{10} * B_4$	$B_{11} * B_4$	Send B_5 to P_2	Idle	$B_{14} * B_5$	t_{20}
$B_8 * B_5$	Compute $B_9 * B_5$ skipped	Recv. B_5 from P_5	Idle	Send B_5 to P_3	Send B_6 to P_7	Recv. B_6 from P_6	t_{21}
Recv. B_6 from P_6	$B_9 * B_5$	Compute $B_3 * B_5$ skipped	Recv. B_6 from P_5	Send B_6 to P_1	Send B_6 to P_1	Compute $B_7 * B_6$	t_{22}
Compute $B_8 * B_6$ skipped	Recv. B_6 from P_6	$B_{10} * B_5$	Compute $B_4 * B_5$ skipped	Compute $B_5 * B_6$	Send B_6 to P_2	$B_{14} * B_6$	t_{23}
$B_8 * B_6$	Compute $B_9 * B_6$ skipped	Recv. B_6 from P_6	$B_{11} * B_5$	$B_{12} * B_6$	Send B_6 to P_3	Idle	t_{24}
Recv. B_7 from P_7	$B_9 * B_6$	Compute $B_3 * B_6$ skipped	Recv. B_6 from P_6	Idle	Send B_6 to P_4	Send B_7 to P_1	t_{25}
Compute $B_8 * B_7$ skipped	Recv. B_7 from P_7	$B_{10} * B_6$	Compute $B_4 * B_6$ skipped	Recv. B_6 from P_6	Send B_6 to P_5	Send B_7 to P_2	t_{26}
$B_8 * B_7$	Compute $B_9 * B_7$ skipped	Recv. B_7 from P_7	$B_{11} * B_6$	Compute $B_5 * B_6$ skipped	Compute $B_6 * B_6$	Send B_7 to P_3	t_{27}
	$B_9 * B_7$	Compute $B_3 * B_7$ skipped	Recv. B_7 from P_7	$B_{12} * B_6$	Compute $B_6 * B_6$	Send B_7 to P_4	t_{28}
		$B_{10} * B_7$	Compute $B_4 * B_7$ skipped	Recv. B_7 from P_7	$B_{13} * B_6$	Send B_7 to P_5	t_{29}
			$B_{11} * B_7$	Compute $B_5 * B_7$ skipped	Recv. B_7 from P_7	Send B_7 to P_6	t_{30}
				$B_{12} * B_7$	Compute $B_6 * B_7$ skipped	Send B_7 to P_7	t_{31}
					$B_{13} * B_7$	Compute $B_7 * B_7$	t_{32}

Figure 3.15: A modified version of the scheme presented in figure 3.14. Now, each processor who has to send a required block, distributes its block to all other processors. First, the processor sends its block to all other processors ahead and then returns to serve the processors behind. Comparison with Fig. 3.14 shows that the idle states of the individual processors are significantly reduced.

its block to all other processors. The time needed to perform the communication is considerably perceptible, particularly, if we have several hundreds of processors. However, the use of as many processors as possible is essential to optimise the computation of the sensitivity matrix. Therefore, an appropriate strategy must be considered to optimise the computation of \mathbf{J} and to minimise the communication in computing $\mathbf{J}\mathbf{J}^T$. Considering the method presented in Fig. 3.4c, we can modify the send phase for the cross product as follows (see also Figs. 3.16 and 3.17):

1. All processors enter two main loops: The first loop runs over all blocks (from 1 to 14 in the example). The second loop runs only over the saved blocks on each processor (2 blocks in this example).

2. While executing the first loop, each processor verifies that it owns the required block. Afterwards, the processor sends this block to two other processors only and enters the computation stage.
3. The two processors which have received the message keep a copy of the block before distributing the block to another four processors. However, each processor sends the block only to two other processors.
4. This scheme can be continued to form a tree of receiving and sending procedures. In the next step, four processors own the block. Similar to the previous step, each processor in this level keeps a copy of the block and then forwards it to another two processors. In total, the four processors in this level forward the block to eight processors (3.17).

The steps above must be continued until all processors have obtained a copy of the block so that they can enter the computation phase, to carry out the multiplication of their own blocks with the received one. Figure 3.17 demonstrates the concept of the send hierarchy used to establish the communication in the method presented in Fig. 3.16. Obviously, none of the processors sends its block to all other processors. Consequently, the required time for the send phase is considerably reduced for each processor, and we can use as many processors as we wish.

In the previous examples, computation tasks marked with yellow boxes in Figs. 3.14, 3.15 and 3.16 can be skipped because of the symmetrical properties of the final result. When considering the computation task of each processor, it becomes evident that some processors accomplish more computational tasks than others. However, considering the presented concept of the modified version of the cross product sample code in Fig. 3.9b, one can further optimise the distribution of the computation tasks. Figure 3.18 demonstrates the scheme in which each processor accomplishes an equal number of block multiplications.

It is interesting to note that the cross product multiplication in the model space ($\mathbf{J}^T\mathbf{J}$) can be computed more efficiently than $\mathbf{J}\mathbf{J}^T$ because fewer communications are required between processors. Each processor keeps one or more blocks in its memory and can compute all column by column multiplications using its own blocks, thereby creating partial sums (sum_local in Fig. 3.19a). After the workers have computed their partial sums, the master collects all local sums to compute the global sum and save it as a vector containing one half of $\mathbf{J}^T\mathbf{J}$ (Fig. 3.19c). Fortunately, this kind of operations is supported by MPI with the subroutine `MPI_REDUCE(local_value, global_value, 1, ..., MPI_SUM, WHERE, ...)` which is part of the collective communication library. The subroutine `MPI_REDUCE` performs a global reduce operation (such as SUM, MAX, LOGICAL, etc.) across all processors located in the same communicator using distributed values. In our examples, the reduce operation is the summation applied to the distributed values of sum_local to compute sum_global on the master

P ₁ owns Blocks: B ₁ , B ₈	P ₂ owns Blocks: B ₂ , B ₉	P ₃ owns Blocks: B ₃ , B ₁₀	P ₄ Owns Blocks: B ₄ , B ₁₁	P ₅ owns Blocks: B ₅ , B ₁₂	P ₆ owns Blocks: B ₆ , B ₁₃	P ₇ owns Blocks: B ₇ , B ₁₄	
Send B ₁ to P ₂	Recv. B ₁ from P ₁	Idle	Idle	Idle	Idle	Idle	t ₁
Send B ₁ to P ₃	Send B ₁ to P ₂	Recv. B ₁ from P ₁	Recv. B ₁ from P ₃	Idle	Idle	Idle	t ₂
Compute B ₁ *B ₁	Send B ₁ to P ₅	Send B ₁ to P ₆	Compute B ₄ *B ₁	Recv. B ₁ from P ₂	Recv. B ₁ from P ₃	Recv. B ₁ from P ₃	t ₃
Idle	Compute B ₂ *B ₁	Send B ₁ to P ₇	Compute B ₄ *B ₁	Compute B ₅ *B ₁	Compute B ₆ *B ₁	Compute B ₇ *B ₁	t ₄
Recv. B ₂ from P ₂	Send B ₂ to P ₁	Compute B ₃ *B ₁	Idle	Compute B ₅ *B ₁	Compute B ₆ *B ₁	Compute B ₇ *B ₁	t ₅
Send B ₂ to P ₄	Send B ₂ to P ₃	Recv. B ₂ from P ₂	Recv. B ₂ from P ₁	Idle	Idle	Idle	t ₆
Send B ₂ to P ₅	Compute B ₂ *B ₂	Send B ₂ to P ₆	Compute B ₄ *B ₂	Recv. B ₂ from P ₁	Recv. B ₂ from P ₃	Idle	t ₇
Compute B ₁ *B ₂ skipped	B ₉ *B ₂	Send B ₂ to P ₇	Compute B ₄ *B ₂	Compute B ₅ *B ₂	Compute B ₆ *B ₂	Recv. B ₂ from P ₁	t ₈
Idle	Idle	Compute B ₃ *B ₂	Idle	Compute B ₅ *B ₂	Compute B ₆ *B ₂	Compute B ₇ *B ₂	t ₉
Recv. B ₃ from P ₃	Send B ₃ to P ₁	Send B ₃ to P ₂	Idle	Idle	Idle	Compute B ₇ *B ₂	t ₁₀
Send B ₃ to P ₄	Recv. B ₃ from P ₃	Send B ₃ to P ₂	Recv. B ₃ from P ₁	Recv. B ₃ from P ₁	Recv. B ₃ from P ₂	Compute B ₇ *B ₂	t ₁₁
Send B ₃ to P ₅	Send B ₃ to P ₆	Compute B ₃ *B ₃	Compute B ₄ *B ₃	Compute B ₅ *B ₃	Compute B ₆ *B ₃	Compute B ₇ *B ₃	t ₁₂
Compute B ₁ *B ₃ skipped	Send B ₃ to P ₇	Compute B ₃ *B ₃	Compute B ₄ *B ₃	Compute B ₅ *B ₃	Compute B ₆ *B ₃	Recv. B ₃ from P ₂	t ₁₃
B ₈ *B ₃	Compute B ₂ *B ₃ skipped	Idle	Idle	Compute B ₅ *B ₃	Compute B ₆ *B ₃	Compute B ₇ *B ₃	t ₁₄
Recv. B ₄ from P ₄	B ₉ *B ₃	Send B ₄ to P ₁	Send B ₄ to P ₁	Idle	Idle	Compute B ₇ *B ₃	t ₁₅
Send B ₄ to P ₂	Recv. B ₄ from P ₄	Recv. B ₄ from P ₁	Send B ₄ to P ₂	Recv. B ₄ from P ₁	Recv. B ₄ from P ₂	Compute B ₇ *B ₃	t ₁₆
Send B ₄ to P ₃	Recv. B ₄ from P ₄	Compute B ₃ *B ₄ skipped	Compute B ₄ *B ₄	Recv. B ₄ from P ₁	Recv. B ₄ from P ₂	Compute B ₇ *B ₃	t ₁₇
Compute B ₁ *B ₄ skipped	Send B ₄ to P ₇	Compute B ₃ *B ₄ skipped	Compute B ₄ *B ₄	Compute B ₅ *B ₄	Compute B ₆ *B ₄	Idle	t ₁₈
B ₈ *B ₄	Compute B ₂ *B ₄ skipped	Idle	Idle	Compute B ₅ *B ₄	Compute B ₆ *B ₄	Recv. B ₄ from P ₂	t ₁₉
Recv. B ₅ from P ₅	B ₉ *B ₄	Send B ₅ to P ₁	Send B ₅ to P ₂	Send B ₅ to P ₁	Send B ₅ to P ₂	Compute B ₇ *B ₄	t ₂₀
Send B ₅ to P ₃	Recv. B ₅ from P ₅	Recv. B ₅ from P ₁	Recv. B ₅ from P ₁	Send B ₅ to P ₂	Recv. B ₅ from P ₂	Compute B ₇ *B ₄	t ₂₁
Send B ₅ to P ₄	Send B ₅ to P ₆	Compute B ₃ *B ₅ skipped	Recv. B ₅ from P ₁	Compute B ₅ *B ₅	Recv. B ₅ from P ₂	Compute B ₇ *B ₄	t ₂₂
Compute B ₁ *B ₅ skipped	Send B ₅ to P ₇	Compute B ₃ *B ₅ skipped	Compute B ₄ *B ₅ skipped	Compute B ₅ *B ₅	Compute B ₆ *B ₅	Recv. B ₅ from P ₂	t ₂₃
B ₈ *B ₅	Compute B ₂ *B ₅ skipped	Idle	Idle	Compute B ₅ *B ₅	Compute B ₆ *B ₅	Compute B ₇ *B ₅	t ₂₄
Recv. B ₆ from P ₆	B ₉ *B ₅	Recv. B ₆ from P ₁	Recv. B ₆ from P ₁	Recv. B ₆ from P ₂	Send B ₆ to P ₁	Compute B ₇ *B ₅	t ₂₅
Send B ₆ to P ₃	Recv. B ₆ from P ₆	Recv. B ₆ from P ₁	Recv. B ₆ from P ₁	Recv. B ₆ from P ₂	Send B ₆ to P ₂	Compute B ₇ *B ₅	t ₂₆
Send B ₆ to P ₄	Send B ₆ to P ₅	Compute B ₃ *B ₆ skipped	Recv. B ₆ from P ₁	Recv. B ₆ from P ₂	Compute B ₆ *B ₆	Compute B ₇ *B ₅	t ₂₇
Compute B ₁ *B ₆ skipped	Send B ₆ to P ₇	Compute B ₃ *B ₆ skipped	Compute B ₄ *B ₆ skipped	Compute B ₅ *B ₆ skipped	Compute B ₆ *B ₆	Compute B ₇ *B ₅	t ₂₈
B ₈ *B ₆	Compute B ₂ *B ₆ skipped	Idle	Compute B ₄ *B ₆ skipped	Compute B ₅ *B ₆ skipped	Compute B ₆ *B ₆	Recv. B ₆ from P ₂	t ₂₉
Idle	B ₉ *B ₆	Idle	Idle	Compute B ₅ *B ₆ skipped	Compute B ₆ *B ₆	Compute B ₇ *B ₆	t ₃₀
Recv. B ₇ from P ₇	Idle	Idle	Idle	Idle	Idle	Compute B ₇ *B ₆	t ₃₁
Send B ₇ to P ₂	Recv. B ₇ from P ₇	Recv. B ₇ from P ₁	Idle	Idle	Idle	Send B ₇ to P ₁	t ₃₂
Send B ₇ to P ₃	Send B ₇ to P ₄	Compute B ₃ *B ₇ skipped	Recv. B ₇ from P ₁	Recv. B ₇ from P ₂	Recv. B ₇ from P ₂	Send B ₇ to P ₂	t ₃₃
Send B ₇ to P ₄	Send B ₇ to P ₅	Compute B ₃ *B ₇ skipped	Recv. B ₇ from P ₁	Recv. B ₇ from P ₂	Recv. B ₇ from P ₂	Compute B ₇ *B ₇	t ₃₄
Compute B ₁ *B ₇ skipped	Send B ₇ to P ₆	Compute B ₃ *B ₇ skipped	Compute B ₄ *B ₇ skipped	Compute B ₅ *B ₇ skipped	Recv. B ₇ from P ₂	Compute B ₇ *B ₇	t ₃₅
B ₈ *B ₇	Compute B ₂ *B ₇ skipped	Idle	Compute B ₄ *B ₇ skipped	Compute B ₅ *B ₇ skipped	Compute B ₆ *B ₇ skipped	Compute B ₇ *B ₇	t ₃₆
Idle	B ₉ *B ₇	Idle	Idle	Compute B ₅ *B ₇ skipped	Compute B ₆ *B ₇ skipped	Compute B ₇ *B ₇	t ₃₇
Idle	Idle	Idle	Idle	Compute B ₅ *B ₇ skipped	Compute B ₆ *B ₇ skipped	Compute B ₇ *B ₇	t ₃₈

Run Time

Figure 3.16: An optimal scheme in terms of communication and computation time to compute the cross product $\mathbf{J}\mathbf{J}^T$ on many processors. In this scheme, each processor has to perform one receive task and two send tasks. None of the processors enters in a loop to send its block to all other processors. Even if the idle state of the processors is similar when compare it to the previous methods (Figs. 3.14 and 3.15), the number of send tasks for each processor is considerably smaller. This becomes very important when distributing the computational problem on a big cluster with several hundreds of processors.

processor (Fig. 3.19c).

To describe the scheme to compute $\mathbf{J}^T\mathbf{J}$ in parallel, we make use of the example to demonstrate the multiplication $\mathbf{J}\mathbf{J}^T$. On each processor, three loops (i,j,k) must be implemented (Fig. 3.19b): The first loop (i) runs over all model parameters M while the first column is kept fixed⁷. The second loop (j) also runs over M but starts from i, because of the symmetrical properties of the final result. The third loop (k) runs over all data responses (NResp) multiplied by the number of blocks on each processor (NResp × NBlocks). In this example we have NResp=8, corresponding to the eight real and imaginary parts of the impedance tensor \mathbf{Z} and two blocks (Fig. 3.19).

Obviously, computing the cross product $\mathbf{J}^T\mathbf{J}$ in parallel requires less communication

⁷In column by column multiplication, one column remains fixed and one moves over all other columns.

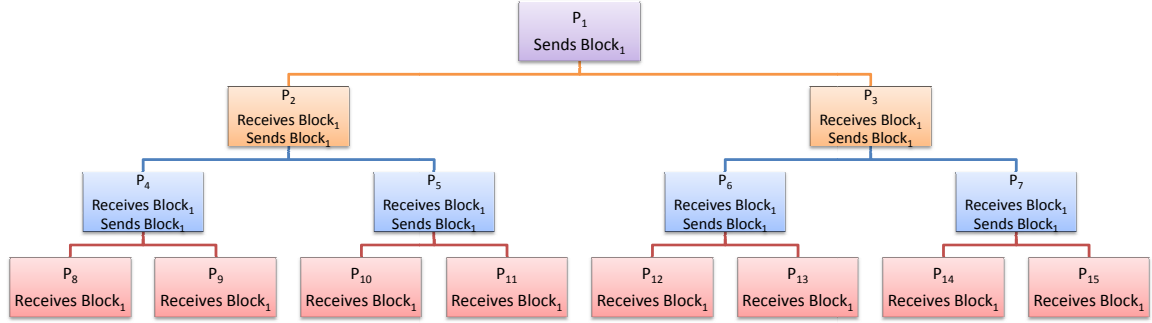


Figure 3.17: The send hierarchy used in figure 3.16. Each processor performs a maximum of one receive task and two send tasks. In the upper most level, processor P_1 has only two send tasks. In the lower most level (P_8 through P_{15}), the processors have only one receive task each.

Processors:	P_1 has: B_1 and B_8	P_2 has: B_2 and B_9	P_3 has: B_3 and B_{10}	P_4 has: B_4 and B_{11}	P_5 has: B_5 and B_{12}	P_6 has: B_6 and B_{13}	P_7 has: B_7 and B_{14}
# of computation tasks:	15	15	15	15	15	15	15

B_1B_1	B_2B_1	B_3B_1	B_4B_1	B_5B_1	B_6B_1	B_7B_1	B_8B_1	X	X	X	X	X	X
X	B_2B_2	B_3B_2	B_4B_2	B_5B_2	B_6B_2	B_7B_2	B_8B_2	B_9B_2	X	X	X	X	X
X	X	B_3B_3	B_4B_3	B_5B_3	B_6B_3	B_7B_3	B_8B_3	B_9B_3	$B_{10}B_3$	X	X	X	X
X	X	X	B_4B_4	B_5B_4	B_6B_4	B_7B_4	B_8B_4	B_9B_4	$B_{10}B_4$	$B_{11}B_4$	X	X	X
X	X	X	X	B_5B_5	B_6B_5	B_7B_5	B_8B_5	B_9B_5	$B_{10}B_5$	$B_{11}B_5$	$B_{12}B_5$	X	X
X	X	X	X	X	B_6B_6	B_7B_6	B_8B_6	B_9B_6	$B_{10}B_6$	$B_{11}B_6$	$B_{12}B_6$	$B_{13}B_6$	X
X	X	X	X	X	X	B_7B_7	B_8B_7	B_9B_7	$B_{10}B_7$	$B_{11}B_7$	$B_{12}B_7$	$B_{13}B_7$	$B_{14}B_7$
X	X	X	X	X	X	X	B_8B_8	B_9B_8	$B_{10}B_8$	$B_{11}B_8$	$B_{12}B_8$	$B_{13}B_8$	$B_{14}B_8$
B_1B_9	X	X	X	X	X	X	X	B_9B_9	$B_{10}B_9$	$B_{11}B_9$	$B_{12}B_9$	$B_{13}B_9$	$B_{14}B_9$
B_1B_{10}	B_2B_{10}	X	X	X	X	X	X	X	$B_{10}B_{10}$	$B_{11}B_{10}$	$B_{12}B_{10}$	$B_{13}B_{10}$	$B_{14}B_{10}$
B_1B_{11}	B_2B_{11}	B_3B_{11}	X	X	X	X	X	X	X	$B_{11}B_{11}$	$B_{12}B_{11}$	$B_{13}B_{11}$	$B_{14}B_{11}$
B_1B_{12}	B_2B_{12}	B_3B_{12}	B_4B_{12}	X	X	X	X	X	X	$B_{11}B_{12}$	$B_{12}B_{12}$	$B_{13}B_{12}$	$B_{14}B_{12}$
B_1B_{13}	B_2B_{13}	B_3B_{13}	B_4B_{13}	B_5B_{13}	X	X	X	X	X	X	$B_{12}B_{13}$	$B_{13}B_{13}$	$B_{14}B_{13}$
B_1B_{14}	B_2B_{14}	B_3B_{14}	B_4B_{14}	B_5B_{14}	B_6B_{14}	X	X	X	X	X	X	$B_{13}B_{14}$	$B_{14}B_{14}$

Figure 3.18: The distribution of the computation tasks among all processors. The matrix shows the final result of the \mathbf{JJ}^T . Each processor is coded with a colour, to distinguish which computation has been carried out on a particular processor. Now, each processor performs the same number of row \times row computation tasks (15).

than \mathbf{JJ}^T . However, the end result (half of $\mathbf{J}^T\mathbf{J}$) with the dimension $M+[(M-1)*M/2]$, where M is the model dimension, requires more memory than storing the half of \mathbf{JJ}^T , which has the dimension $N+[(N-1)*N/2]$, where N is the number of the data points. Consequently, solving the normal equation in the model space on one processor in a serial manner can take a long time. This difficulty of having to solve the normal equation with huge dimensions can be avoided by parallelizing the corresponding solver of linear equations. As pointed out in the introduction of this chapter, solving a linear set of equations in parallel is a great challenge for mathematicians. Several techniques, strategies and methods have been already developed and implemented in ready to use libraries. In libraries such as ScaLAPACK and PLAPACK, which are the parallelized version of the original serial LAPACK library, one can find methods (i.e. Cholesky and LU decomposition) to solve the normal equation in data and in the model space

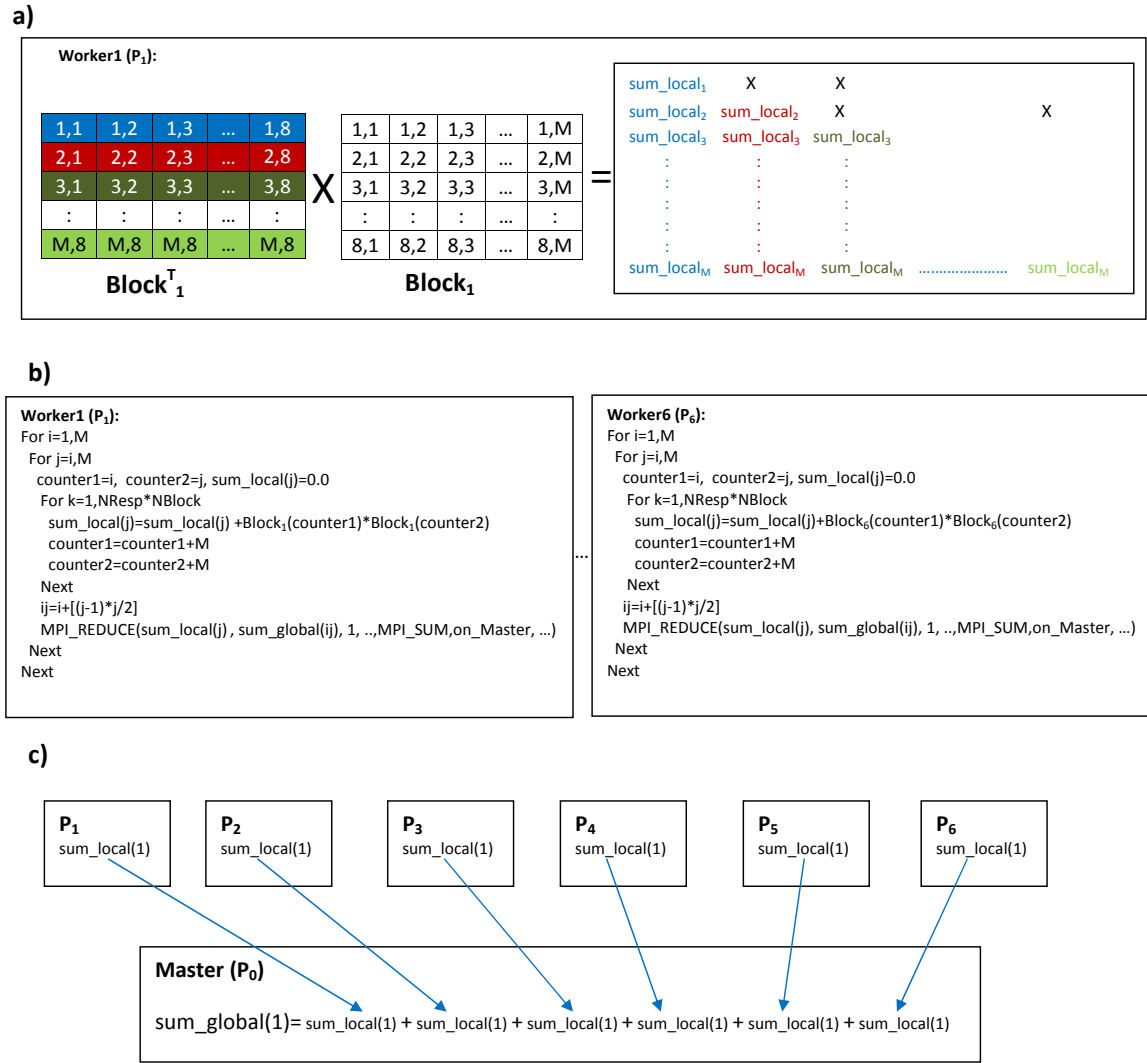


Figure 3.19: The procedure to compute the cross product $\mathbf{J}^T \mathbf{J}$. a) In this case each processor can compute the partial sums “*sum_loc*“ by multiplying each column of a block with all other columns. b) The implementation of the loops *i,j* and *k* is identical for each processor. c) The MPI subroutine `MPI_REDUCE(...)` collects on the master P_0 all local sums “*sum_loc*“ from all worker processors and saves them in the global sum “*sum_global*“.

in parallel. However, all of these libraries require that the equation system is stored on one processor. The distribution of the problem to parallel processors is done internally by the library. This is usually no problem for data space inversion, since the normal equation has the dimension of $N+[(N-1)*N/2]$. However, in the model space, the dimension of the problem may easily exceed the memory available on one processor. But, it is feasible to modify the existing parallelized solvers to solve distributed equation systems without having to store it on one processor. This however is beyond

the scope of this thesis.

3.4.3 Further mathematical operations with the sensitivity matrix in parallel

In the previous subsection, the developed methods to compute the cross products $\mathbf{J}\mathbf{J}^T$ and $\mathbf{J}^T\mathbf{J}$ in parallel have been discussed. However, before calculating these cross products, the sensitivity matrix (\mathbf{J}) must first be smoothed with the model covariance matrix (\mathbf{C}_m) in data space inversion or weighted with the data covariance matrix (\mathbf{C}_d) in the model space inversion (see eqs. 2.60, 2.65, and 2.72). In another words, prior to calculating the cross products $\mathbf{J}\mathbf{J}^T$ and $\mathbf{J}^T\mathbf{J}$, \mathbf{J} must be converted into:

$$\begin{aligned}\tilde{\mathbf{J}} &= \mathbf{J}\mathbf{C}_m^{1/2} && \text{Data space inversion} \\ \tilde{\mathbf{J}} &= \mathbf{C}_d^{-1/2}\mathbf{J} && \text{Model space inversion}\end{aligned}$$

where $\mathbf{C}_m^{1/2}$ and $\mathbf{C}_d^{-1/2}$ are one half of the model smoothing and one half of the data weighting, respectively (Siripunvaraporn & Egbert, 2000; Egbert, 2006). Hence, the actual cross products $\mathbf{J}\mathbf{J}^T$ and $\mathbf{J}^T\mathbf{J}$ for which we discussed their parallelization schemes in the previous section, take the form:

$$\tilde{\mathbf{J}}\tilde{\mathbf{J}}^T = \mathbf{J}\mathbf{C}_m^{1/2}\mathbf{C}_m^{1/2}\mathbf{J}^T = \mathbf{J}\mathbf{C}_m\mathbf{J}^T \quad (3.1)$$

$$\tilde{\mathbf{J}}^T\tilde{\mathbf{J}} = \mathbf{J}^T\mathbf{C}_d^{-1/2}\mathbf{C}_d^{-1/2}\mathbf{J} = \mathbf{J}^T\mathbf{C}_d^{-1}\mathbf{J} \quad (3.2)$$

Starting with eq. 3.1, the regularization term \mathbf{C}_m in data space inversion can be obtained without having to create and save \mathbf{C}_m (Siripunvaraporn *et al.*, 2005). This can be explained if we analyse the effect of \mathbf{C}_m on \mathbf{J}^T . In general, \mathbf{C}_m causes a smoothing of the updated model parameters after solving the normal equation (in data or model space). However, in the formulation of data space inversion, \mathbf{C}_m acts on the sensitivity matrix to generate smoothed rows of \mathbf{J} , and consequently smoothed model parameters. For parallelization schemes of the cross product $\mathbf{J}\mathbf{C}_m\mathbf{J}^T$, it is very important to know how to overcome the difficulty of having to create the smoothed rows of \mathbf{J} without having to calculate the multiplications $\mathbf{J}\mathbf{C}_m^{1/2}$ and $\mathbf{C}_m^{1/2}\mathbf{J}^T$ with an implicit use of $\mathbf{C}_m^{1/2}$. Siripunvaraporn & Egbert (2000) showed that one row of \mathbf{J} (Note that the length of one row of \mathbf{J} is equal to M , the total number of model parameters) can be smoothed by convolution with \mathbf{C}_m . The convolution process is achieved numerically by solving a 2D (3D) diffusion equation in 2D (3D) case (Further details on this approach can be found in Siripunvaraporn & Egbert (2000)).

For the parallelization scheme of the cross product $\mathbf{J}\mathbf{C}_m^{1/2}\mathbf{C}_m^{1/2}\mathbf{J}^T$, it is very important to notice that the convolution affects on each row of \mathbf{J} , however, independently. Since the "rough" rows of \mathbf{J} are calculated independently, we can apply the convolution of each "rough" row with \mathbf{C}_m to yield a smoothed row. This is achieved immediately after computing one row in the computation stage of \mathbf{J} . Consequently, after the

computation stage of \mathbf{J} and prior to the cross product, smoothed blocks of \mathbf{J} are available on each processor (Fig. 3.20).

The multiplication of \mathbf{J}^T with \mathbf{C}_d^{-1} in the model space (eq.) also can be obtained at the computation stage of \mathbf{J} . Noting that the matrix \mathbf{C}_d^{-1} , which contains the inverse of the data error (see eq. 2.55) is diagonal, the multiplication can be done by multiplying one particular element in each row of \mathbf{J} with the corresponding value in \mathbf{C}_d^{-1} .

In both data and model space inversion, the right hand side of the normal equations (2.72 and 2.106) contains an expression that depends on the multiplication of \mathbf{J} with the model parameters (\mathbf{m}):

$$\hat{\mathbf{d}}_n = \mathbf{d} - \mathbf{F}(\mathbf{m}_n) + \mathbf{J}_n \mathbf{m}_n \quad (3.3)$$

where \mathbf{d} represents the data vector, $\mathbf{F}(\mathbf{m})$ the model response vector, \mathbf{m}_k the model parameters at the n 'th iteration. The multiplication $\mathbf{J}_n \mathbf{m}_n$ can be done very efficiently when considering that each row of \mathbf{J} must be multiplied with \mathbf{m}_n . Since \mathbf{m}_n is broadcasted from the master to all processors to compute $\mathbf{F}(\mathbf{m}_n)$, the multiplication $\mathbf{J}_n \mathbf{m}_n$ can be executed in parallel while computing each element of \mathbf{J}_n .

We can extend this approach to compute $\hat{\mathbf{d}}_k$, because \mathbf{d} is stored on all processors and $\mathbf{F}(\mathbf{m}_k)$ for one particular period is also available on the current processor. However, proper bookkeeping of all indices of received periods and stations is very important to make sure that equation 3.3 is implemented correctly.

Figure 3.20 demonstrates the computation of \mathbf{J} as shown in Fig. 3.12 with emphasis on the implemented steps to compute $\hat{\mathbf{d}}_k$ and to smooth the rows of \mathbf{J} .

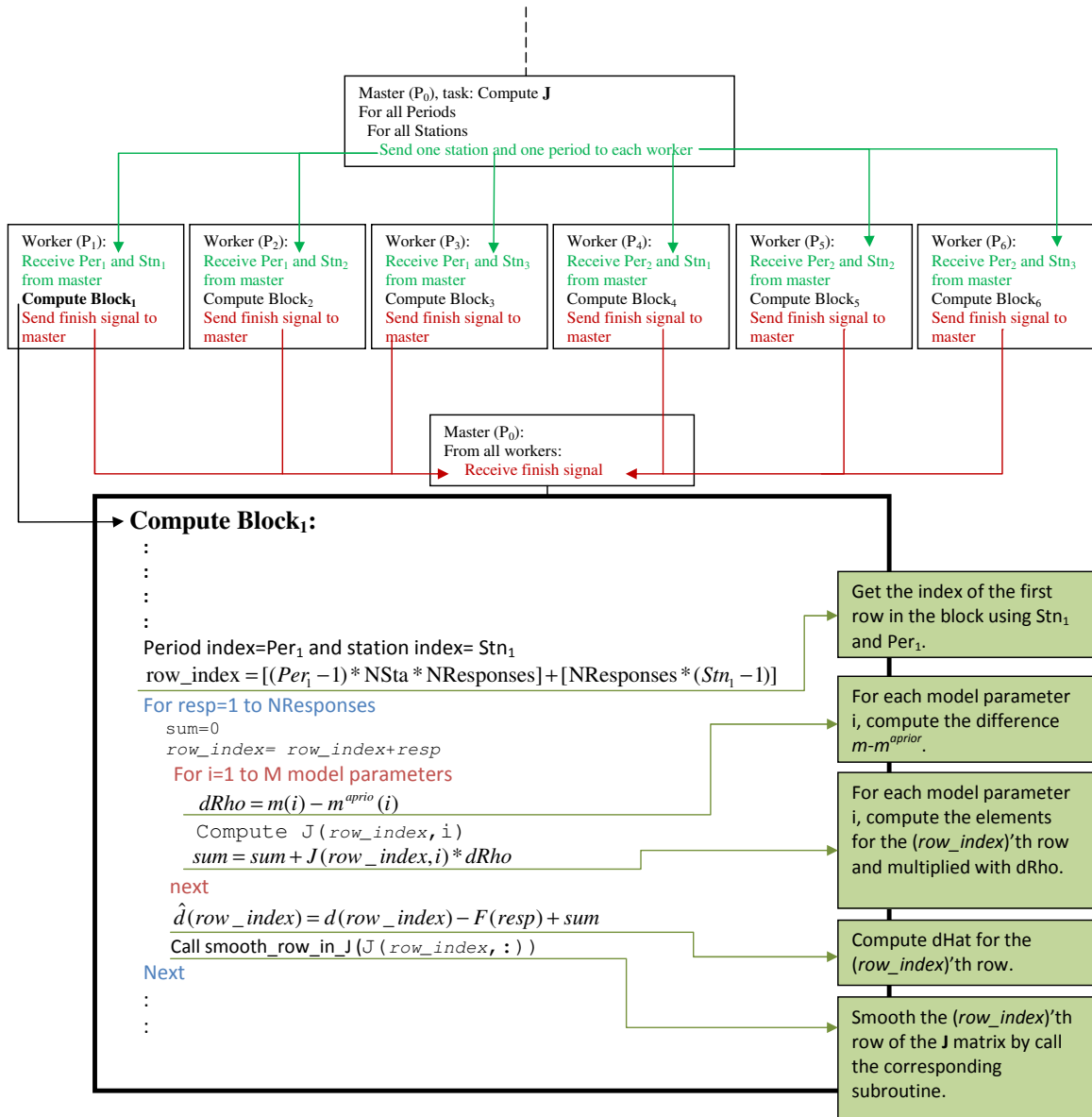


Figure 3.20: Part of the code inside routine "Compute $Block_i$ " (see figure 3.12) to highlight the steps required to compute $\hat{\mathbf{d}}_k$ and a smoothed sensitivity matrix (\mathbf{J}).

3.5 Generality and performance tests of the parallel scheme using synthetic data

The previously developed parallel schemes were written independently from the codes that solve the forward modeling and inversion problems. Therefore, these parallel schemes can be considered as generic schemes that can be linked to any source code to convert it from serial to parallel with minimal changes in the original code. Figure 3.21 clarifies this idea in the context of solving the forward modeling problem.

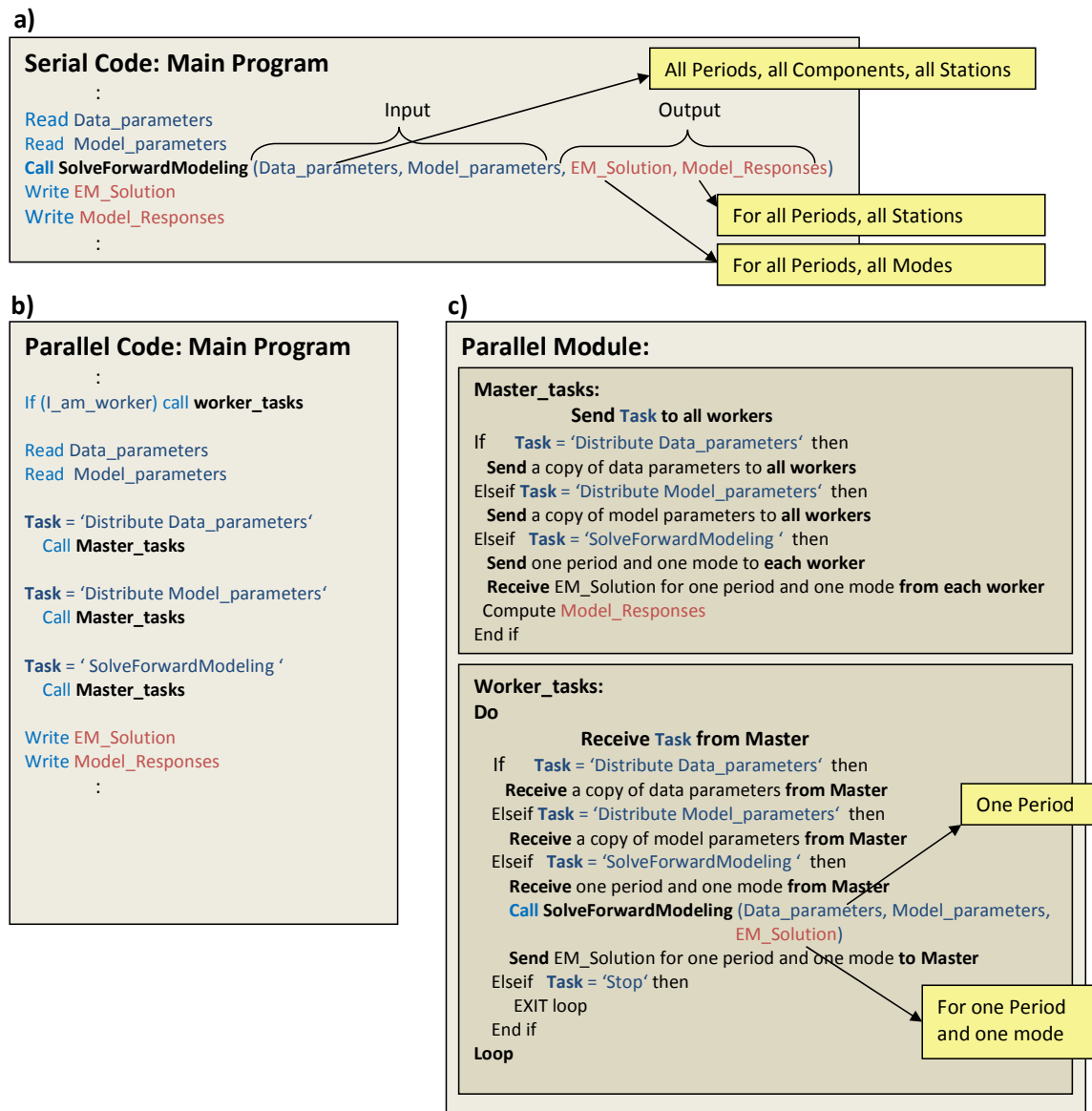


Figure 3.21: Rewriting the main program from serial to parallel. a) The serial code that must be parallelized which contains the instruction 'Call SolveForwardModeling'. b) The parallelized version linked to the parallel module which contains the instructions 'Master_tasks' and 'Worker_tasks'. c) The parallel module which contains all necessary instructions to parallelize the instruction 'Call SolveForwardModeling'.

The serial version in Fig. 3.21a contains the instruction 'Call SolveForwardModeling' which solves the forward modelling problem. Inputs for this instruction are the data and model parameters. The output is the electromagnetic solution (EM_solution) and the model responses. The single instruction ('Call SolveForwardModeling') multi data (several periods and modes included in data parameters) architecture (SIMD) presented in Fig. 3.21a can be parallelized if we link it to the developed parallel

module (Fig. 3.21c). In this case, the serial main program is rewritten as presented in Fig. 3.21b to establish a link to the parallel module. After initializing the MPI parameters, the parallel main program starts with sending all workers to a queue ('Call Worker_tasks'). All workers enter in this queue, which is simply an infinite loop, to wait for a message from the master indicating the next task ('Receive Task from Master'). The master reads the instructions in the main program. The first task for the master is reading the input parameters (data and model). After reading the input parameters, the master, who is the only processor that owns the input parameters, starts distributing the parameters to all workers, so that each worker obtains a copy of the data and model parameters. Setting the variable 'Task' to 'Distribute Data_parameters' followed by calling the instruction 'Call Master_tasks' in the main program, lets the master enter the parallel module. After entering the parallel module, in particularly the 'Master_Tasks', the sends the 'Task' message to all workers, which is in this case 'Distribute Data_parameters'. The workers who are waiting in the queue receive the 'Task' message from the master and go directly to the position in the 'Worker_tasks' where they can receive a copy of the data parameters from the master. Once the master has finished sending a copy of the data parameters, it returns to the main program to read the next task. After having received the data from the master, the workers go back to the top of the queue and wait for a new task. The same steps are performed when executing other tasks, the existing tasks included in the parallel module are listed in table 3.2. The tasks listed in table 3.2 cover the most important and time consuming steps in solving the forward modeling and inversion problems. These tasks can be nested in any serial code to convert it to parallel code. However, minor modifications are necessary in the parallel module to adjust input/output for each instruction in the serial code.

The generality of the parallel module has been tested by nesting it in two serial codes that we have access to: WS3DMTINV and the Modular System.

WS3DMTINV is a serial code developed by Siripunvaraporn (Siripunvaraporn *et al.*, 2005) to invert MT data in 3D. The inversion scheme used in this code is Occam's inversion solved in data space. The forward modeling problem is solved on a staggered 3D grid by using the finite difference method (FD). The complex linear equation system resulting from applying FD (eq. 2.49) is then solved iteratively using the quasi minimum residual (QMR) method with incomplete LU-decomposition as preconditioner. On the other hand, the real normal equation resulting from using occam inversion in data space (eq. 2.106) is solved by Cholesky decomposition.

The Modular System developed by Egbert (Egbert, 2006) is also a serial code, however, it was written to invert electromagnetic data in multi dimensions (2D and 3D). Similar approaches as in WS3DMTINV were used in the Modular System to solve the MT (and any other EM method) forward modelling problem. The inversion module included in the Modular System is considered as being as generic as possible. Thus, it

Task	Description
Distribute_Data_Parameters	Sending a copy of the data parameters to all workers.
Distribute_Model_Parameters	Sending a copy of the model parameters to all workers.
SolveForwardModeling	Solving the forward modelling problem with the electromagnetic solution as an optional output.
SolveForwardModeling_INV	Solving the forward modelling problem with the electromagnetic solution kept on each processor. Used when running the inversion to compute the sensitivity matrix.
Compute_Sens_Matrix	Computing the sensitivity matrix in parallel.
Computing the cross product $\mathbf{J}\mathbf{J}^T$	
Cross_product_JJT_one_proces	On one processor.
Cross_product_JJT_method1	Using the method presented in Fig. 3.14.
Cross_product_JJT_method2	Using the method presented in Fig. 3.15
Cross_product_JJT_method3	Using the method presented in Fig. 3.16.
Cross_product_JTJ	Computing the cross product multiplication $\mathbf{J}^T\mathbf{J}$
Model_update	Computing $\mathbf{C}_m\mathbf{J}^T\beta$. Used when solving the inverse problem in data space using Occam inversion scheme.
Compute_JT_mult_d	Computing the product $\mathbf{J}^T\mathbf{d}$. Used when applying the non-linear conjugate gradient (NLCG) method to solve the inverse problem.
Compute_J_mult_m	Computing the product $\mathbf{J}\mathbf{m}$. Used when applying the conjugate gradient method to solve the normal equation.

Table 3.2: The tasks included in the parallel module together with their description.

does not depend on the type of the electromagnetic data (MT, CSEM, CSMT, etc.) nor on the dimension of the problem (2D, 3D). At present, the inversion module in the Modular System contains the non linear conjugate gradient (NLCG) inversion scheme to invert MT data in 2D and 3D. As explained in section 2.6, the NLCG inversion scheme does not require an explicit use of the sensitivity matrix \mathbf{J} . However, multiplication like $\mathbf{J}^T\mathbf{d}$ is essential in NLCG. The developed parallel scheme covers also this multiplication (see table 3.2).

The parallel module was linked to the two serial codes mentioned above to verify the speed-up factor, first in solving the forward modeling problem and then in solving the inverse problem in 3D. Furthermore, I evaluated the memory requirements for computing the sensitivity matrix using the memory-efficient concept presented in section 3.4.1).

The first 3D model under consideration consists of a resistive ($1000 \Omega m$) and a conductive ($10 \Omega m$) block, embedded within a $100 \Omega m$ half space (Fig. 3.22). The vertical discontinuity between the blocks is located at the midpoint of the grid (point

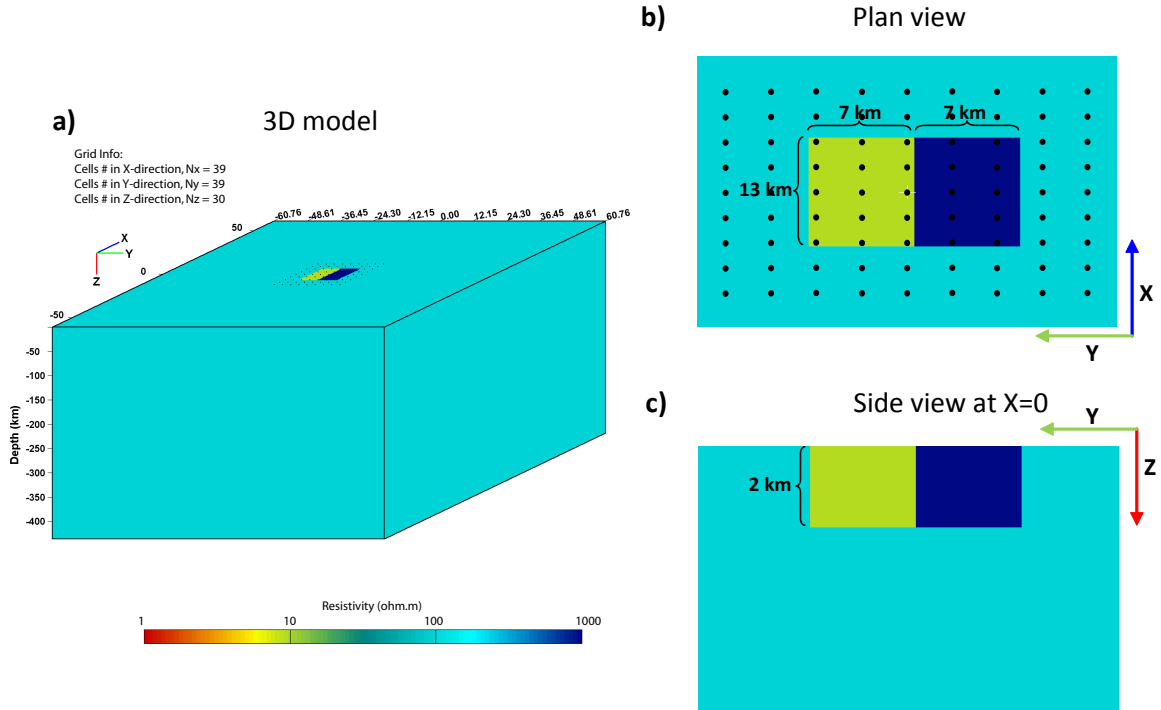


Figure 3.22: A 3D model used to test the speed of the developed parallel scheme for solving the MT forward modelling problem. a), b) and c) presenting the 3D model from different perspectives.

0,0,0). Each block has the dimensions of $13 \times 7 \times 2$ km in the X, Y and Z directions, respectively. The top of the blocks is located at the earth's surface ($z=0$) (Figs. 3.22b and c). Nine profiles with 24 km length each extending from west to east were created. On each profile there are 9 stations with 3 km distance between stations (black dots in Fig. 3.22b). The distance between two profiles is 3 km. In total there are 81 stations for which the model responses must be calculated. Applying the FD method to solve the forward modeling problem and therefore computing the model responses requires (as discussed in section 2.5) discretizing the model into cubes. The 3D model is discretized into 39 cubes in both X and Y directions and 30 layers in Z direction plus 7 air layers above the earth surface ($N_x=N_y=39$, $N_z=30+7$). Thus, in total there are $39 \times 39 \times 37 = 56277$ model parameters used to simulate the distribution of the electromagnetic field in the model and finally computing the model responses at the desired stations location. The used period range extends from 1000Hz to 1000s over 6 decades with 3 periods per decade, 19 periods in total. Since both WSINV3DMT and the Modular system use the same approach to solve the forward modeling problem, either of them can be used to perform a comparison between the serial version and the parallelized version. To test the speed-up factor, the same model and data parameters were used with different numbers of processors. Figure 3.23 demonstrates both

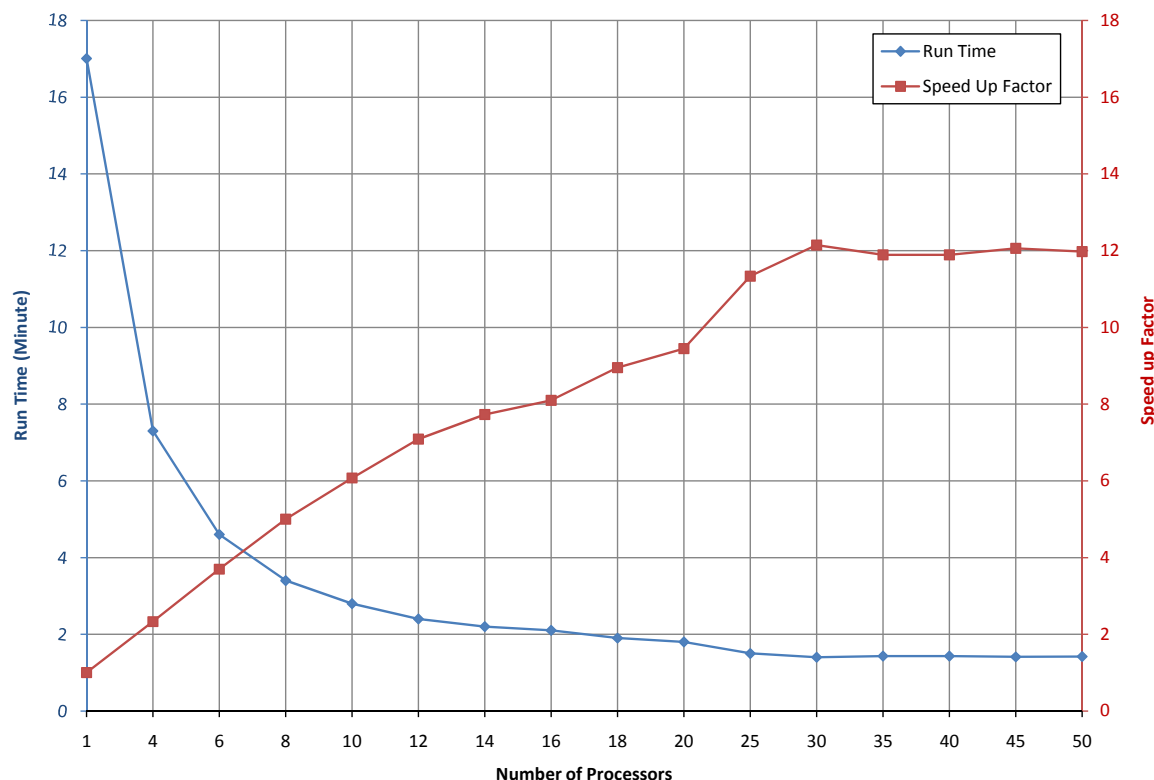


Figure 3.23: Runtime and speed-up factor as a function of number of processors for solving the 3D MT forward modelling problem. The result shows that using only 30 processors is enough to reach a maximum speed-up factor of 12 (red curve).

the run time required to compute the model responses (blue curve) and the speed up factor (red curve).

As expected, the run time decreases with increasing number of processors. However, the results presented in figure 3.23 show that using only 30 processors is enough to reach the minimum run time (1.4 minutes) and the maximum speed up factor of 12. Even when using 38 processors, so that each processor can solve the problem for one period and mode, the required run time remains the same (around 1.4 minutes). This result is not surprising, especially if we examine the time required for solving the forward modeling problem for each period and mode (Fig. 3.24): The convergence of the QMR method to the desired accuracy depends on the used period and the model structures. For periods between 0.1s and 10s, the number of QMR iterations reaches its maximum (356 iterations) at the period of 1s and the first mode for the model under consideration (Fig. 3.24, dark blue curve). These 350 iterations require 1.43 minutes of runtime on the computer (Fig. 3.24, light blue curve).

Even if each processor works on one period and one mode, the processor working with the period of 1s and first mode needs 1.43 minutes to finish its computation.

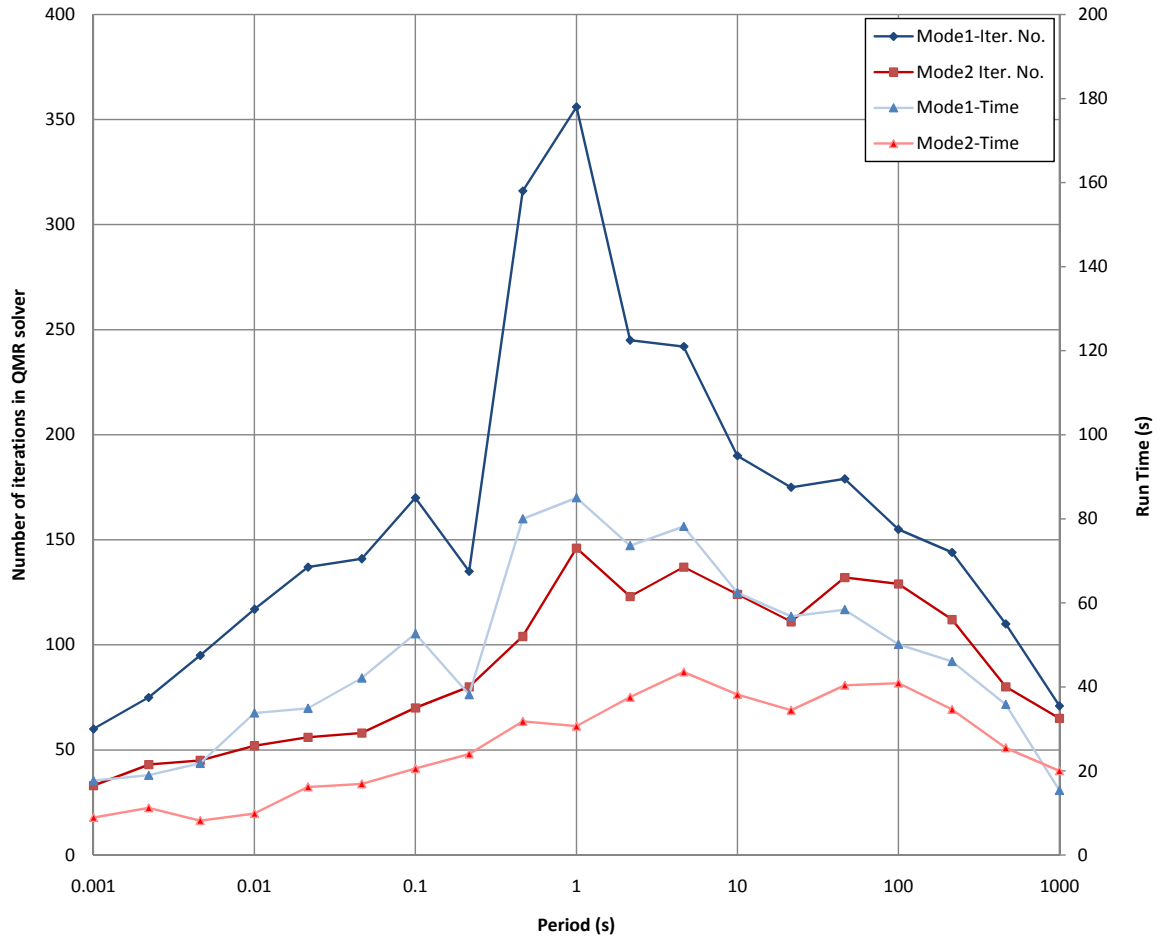


Figure 3.24: Number of QMR iterations (dark blue and red curves) and the run time (light blue and red curves) as a function of periods. At 1s and the first mode, the QMR requires ~ 350 iteration to reach the desired error in solving eq. 2.52 for the EM field (dark blue curve). These ~ 350 iterations require 1.43 minutes runtime on the machine (light blue curve; the light blue and red curves are associated with the time axis at the right hand side).

Consequently, the master must wait for at least 1.43 minutes to get a ready signal from the processor that is working with that particular period to finish the forward modelling computation. This means that, if we are only interested in solving the forward modelling problem, 30 processors are enough to optimise this task. However, if solving the forward modelling is part of the inversion process, we are interested in using as many processors as we can to achieve the task 'Compute_J' (see table 3.2). This task is used when running WSINV3DMT to solve the 3D MT inverse problem. The following inversion test will clarify this issue.

For the 3D inversion tests, the model responses computed in the previous 3D forward modelling test will be used. However, 2% error are added to the computed responses to simulate real data. Starting with Occam inversion in data space used in WSINV3DMT code, this inversion scheme requires explicit use of the sensitivity matrix \mathbf{J} . Thus, directly after solving the forward modelling problem, which is now part of the inversion process, the entries of \mathbf{J} must be computed and saved. Computing and saving the entries of \mathbf{J} is done block wise that distributed over all processors (see subsection 3.4.1). For the data set under consideration, there are 1539 blocks (81 stations \times 19 periods), since each block contains the entries of \mathbf{J} corresponding to one period and one station. The results presented in Fig. 3.23 suggest that using only 30 processors is sufficient to accomplish the forward modelling task. However, using only 30 processors in computing and saving the entries of \mathbf{J} causes the following problems:

- The computation of 1539 blocks will be distributed among 30 processors. This means that each processor will compute 51 blocks, in average, out of 1539 blocks. This computation is not efficient in terms of the run time.
- Furthermore, the memory required to save 51 blocks will be huge, if we consider the number of model parameters and data responses involved in the inversion. In the example under consideration, there are 56277 model parameters (39 $N_x \times$ 39 $N_y \times$ 37 N_z) and eight responses at each station and period representing the real and imaginary parts of the four impedance tensor elements Z_{xx} , Z_{xy} , Z_{yx} and Z_{yy} . Thus, each block contains 450216 (56277×8) elements and the 51 blocks on one processor contain 229661016 elements and require 140 MB⁸ of memory.

The 3D inversion starts with a homogeneous half space (50 Ωm) using the model presented in Fig. 3.22. The distributed 81 stations and 19 periods at each station were used for the 3D inversion. Six iterations were required to fit the full impedance tensor components and reach an RMS value of 0.95.

The 3D model presented in Fig. 3.25 shows that the top and the bottom of the blocks

⁸Each element requires 8 byte on the machine, if it is saved as a real number.

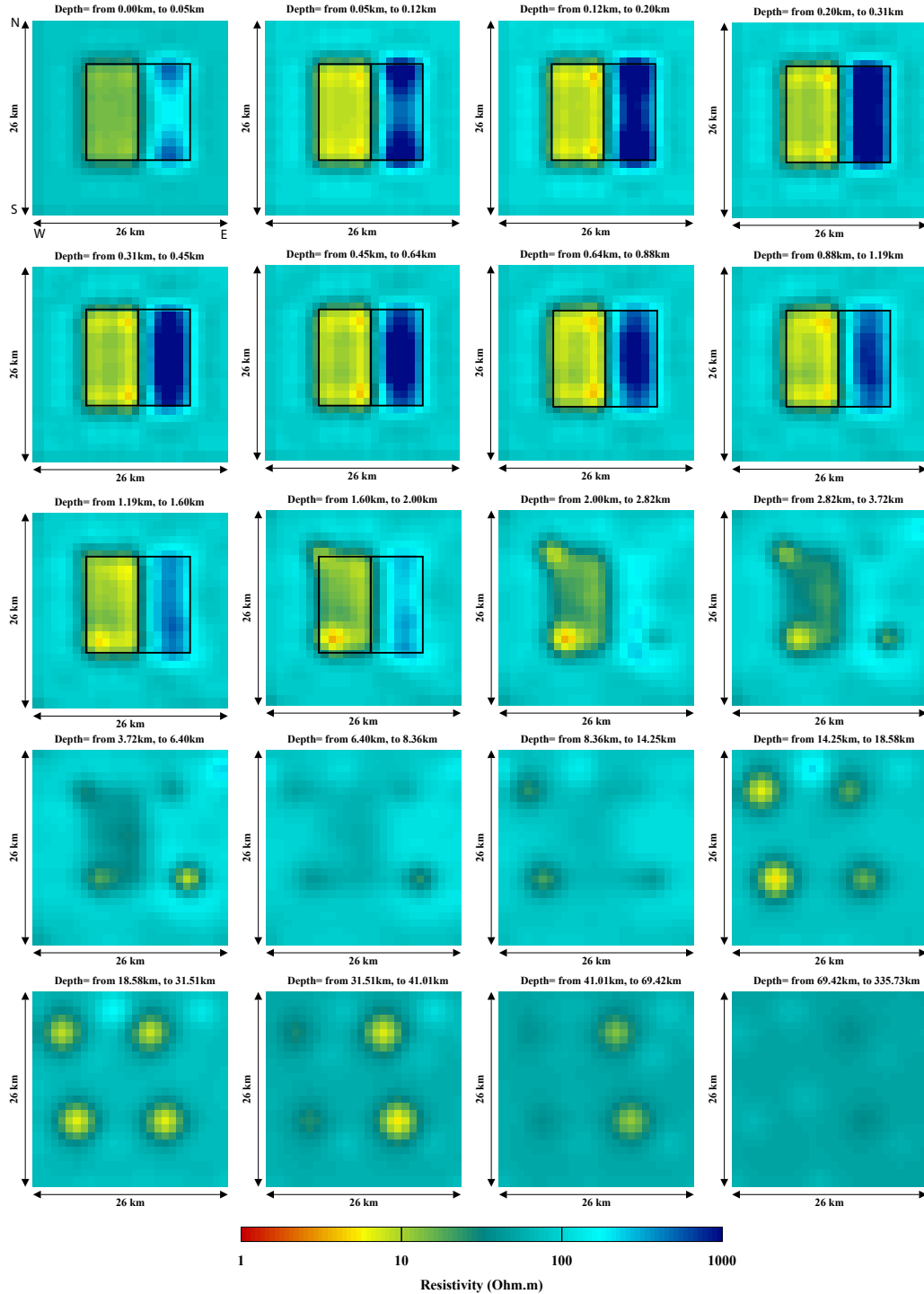


Figure 3.25: The 3D inversion result after 6 iterations with an RMS of 0.95 presented as a plane view slices. The rectangles indicate the position of the original conductive and resistive blocks (see Fig. 3.22) to generate the synthetic data. The result shows that in contrast to the conductive block, the top and the bottom of the resistive are not well resolved. Furthermore, four conductive blobs appear in the model at depths greater than 4 km located at the western and the eastern corners of the conductive and the resistive blocks, respectively.

are not well resolved. Furthermore, the result shows that the bottom of the conductive block smears to a depth of ~ 3 km. Moreover, four conductive blobs appear in the model at depths from ~ 4 km to ~ 50 km. The locations of these blobs coincide with the western and the eastern corners of the conductive and the resistive blocks, respectively.

Aside from the accuracy of the result obtained from the 3D inversion, we are interested in this section to measure the speed of the parallelized version of the code. The 3D inversion starts using a minimum number of 4 and a maximum number of 250 processors. Due to the fact that the serial code requires at least 6 GB to save only the sensitivity matrix, it was impossible to run the serial code for this particular problem using the available hardware. Moreover, using more than 250 processors was difficult because of restrictions on the cluster where I solved the problem. However, increasing the processors number from 4 to 250 gives an idea about the efficiency and the speed of the parallel code.

Since various mathematical operations are involved in the data space occam inversion scheme (as discussed above), the speed of each operation as a function of the processor numbers is analysed. This give us a better idea about the performance of each operation.

Figure 3.26 shows the run time curves for solving the forward modeling problem (a), computing the sensitivity matrix (b), achieving the cross product (c) and the total run time for the 6 iterations (d). Figure 3.26a shows (as discussed above along with Fig. 3.23) that using more than 30 processors in solving the forward modelling problem does not decrease the run time. However, a run time reduction from 12 minutes when using 4 processors to 2.5 minutes when using 30 processors and more is achieved. The need to use many more processors than required to solve the forward modelling problem for the 3D inversion is obvious when considering the run time required to compute the entires of the sensitivity matrix. Figure 3.26b shows that using only 4 processors to compute the sensitivity matrix requires ~ 10 hours for each iteration. A first run time reduction from ~ 10 hours to ~ 50 minutes is reached when using 35 processors to compute the sensitivity matrix. Another run time reduction from ~ 50 minutes to ~ 11 minutes can be observed in Fig. 3.26b when using more than 150 processors . Due to the heterogeneity of the used cluster⁹, Fig. 3.26b shows that the run times required to compute the sensitivity matrix using 40, 45 and 50 processors are considerably longer than when using 35 processors. This effect can also be observed when comparing the run times between using 125 and 150 processors.

The next mathematical operation after computing the sensitivity matrix towards the model update is achieving the cross product $\mathbf{J}\mathbf{C}_m\mathbf{J}^T$. As discussed above, this operation is time consuming and requires many communications between the processors

⁹ Different kind of processors.

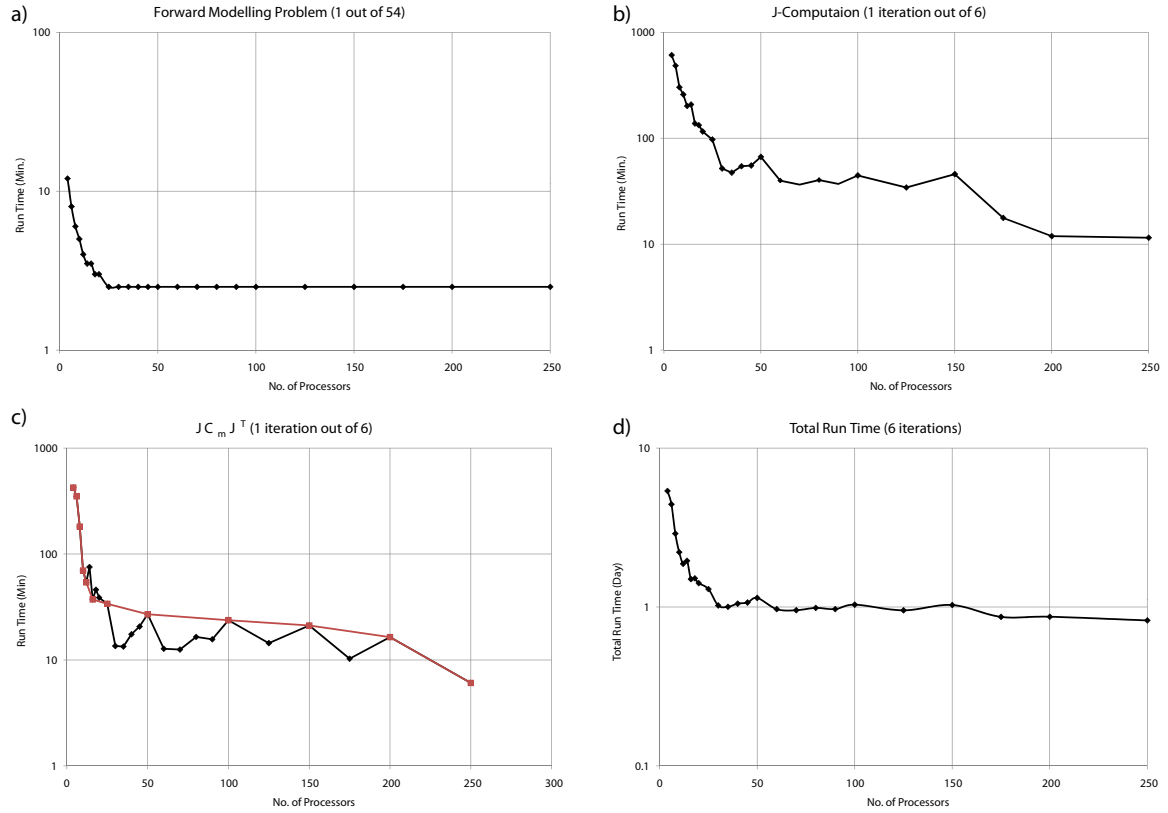


Figure 3.26: The run time curves for a) solving the forward modelling problem, b) computing the sensitivity matrix, c) computing the cross product and d) the total run time to achieve 6 iterations, as function of processors. The red curve in c) presents the run times at certain numbers of processors. It shows the general time reduction in computing the cross product.

to exchange blocks of the sensitivity matrix. Figure 3.26c (black curve) shows that the run time curve for this operation using different number of processors is unstable (increases and decreases for various processors number). This behaviour of the run time curve in Fig. 3.26c reflects the heterogeneity of the used cluster and the traffic loading of the network connection. Nevertheless, a run time reduction from ~ 400 minutes when using 4 processors to ~ 13 minutes when using 35 processors can be observed in the run time curve in Fig. 3.26c. The run time between processor numbers 35 and 200 varies between ~ 25 and ~ 10 minutes per iteration. Using 250 processors to achieve the multiplication $\mathbf{J}\mathbf{C}_m\mathbf{J}^T$ requires only ~ 6 minutes. In general, the run time curve in Fig. 3.26c (black curve) shows a decreasing trend when considering the run times at particular processor numbers (red curve in Fig. 3.26c).

Considering now the total run time to achieve the 6 iterations for the 3D inversion (Fig. 3.26d), a run time decreasing from ~ 5 days when using 4 processors to less than 1 day when using 250 processors is reached. The first run time reduction in Fig. 3.26d is observed when using 35 processors. From processors number 35 to 150, a

roughly constant total run time of ~ 1 day can be observed in Fig. 3.26d. The use of more than 150 processors to achieve the 6 iteration decreases the total run time to less than 1 day.

The second serial code that has been linked to the parallel module I have developed in the frame work of my thesis is the Modular system. The main inversion scheme implemented in this code is the NLCG scheme (Non linear conjugate gradients; Kelbert, pers. comm.). As pointed out in subsection 2.6.4, the NLCG scheme does not require an explicit use of the sensitivity matrix and consequently there is no need to compute $\mathbf{J}\mathbf{C}_m\mathbf{J}^T$ or $\mathbf{J}^T\mathbf{C}_m^{-1}\mathbf{J}$. The only mathematical operations required for the NLCG scheme are the solution of the forward modelling problem and computing the gradient $\mathbf{J}^T\hat{\mathbf{d}}$, where $\hat{\mathbf{d}}$ is the data residual (measured - predicted data). The latter operation is already parallelized and included in the parallel module (see table 3.2). However, the parallelization scheme that has been linked to the Modular system runs only over periods when solving the forward modelling problem. A parallelization over periods and modes was currently not possible because of technical difficulties in keeping the original serial code in its generic form.

The same synthetic data set and the same discretization of the model presented in Fig. 3.22 are used for the 3D inversion in parallel and serial in the Modular system. A 2% error was added to all impedance tensor values (Z_{xx} , Z_{xy} , Z_{yx} and Z_{yy}) to be inverted. A homogeneous model ($50 \Omega m$) is used as a starting model. The same number of stations (81 stations) and periods (19 periods) were used. Since the parallelization scheme for the Modular system runs only over periods, the maximum number of processors that can be used for this example is 20. Figure 3.27 shows the 3D inversion result after 50 iterations with an RMS value of 1.02.

In comparison to the 3D model obtained from the data space occam inversion (Fig. 3.25), the model obtained from the NLCG inversion seems to be rough even when using the same computation and parameters of the model covariance matrices (\mathbf{C}_m) in both algorithms. However, the top and bottom of the resistive block are better resolved in the 3D model of Fig. 3.27. Moreover, the four deep conductive blobs appearing in the 3D model presented in Fig. 3.25 do not appear in the 3D model presented in Fig. 3.27. The difference between the number of iterations required for the data space occam inversion (6 iterations) and the NLCG (50 iterations) reflects the non-quadratic behaviour of the NLCG to converge to the desired error.

A comparison between the results obtained from both algorithms is out of scope of my thesis. What we are interested in is the speed of the NLCG inversion algorithm with respect to convergence to the desired error level. As before, the most time consuming mathematical operations used in the NLCG algorithm are analysed individually in terms of the run time.

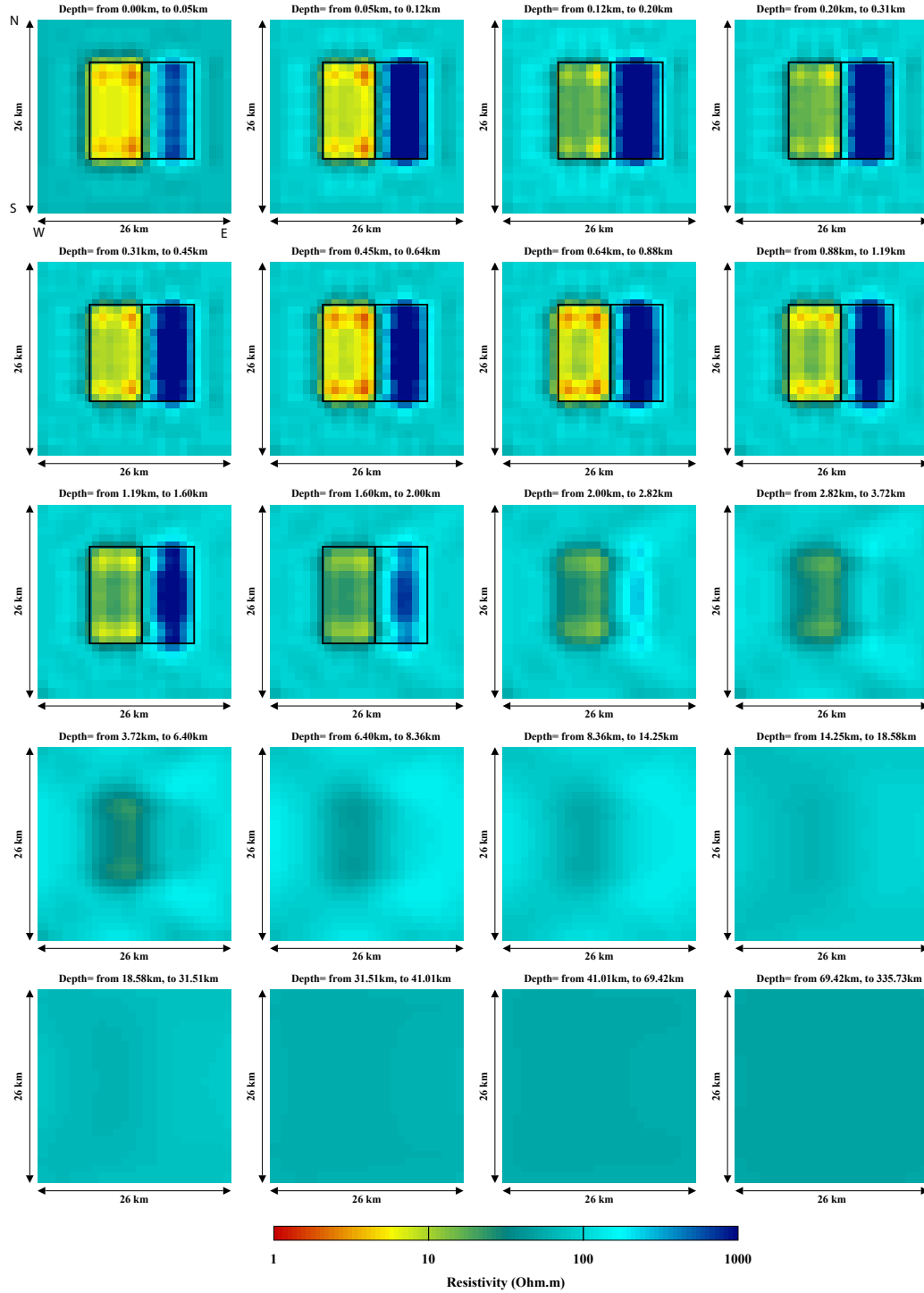


Figure 3.27: The 3D inversion result after 50 iterations with an RMS of 1.02 using the Modular system. The rectangles indicate the position of the original conductive and resistive blocks (see Fig. 3.22). The result indicates that the top and bottom of the resistive block is better resolved. However, the model seems to be rough in comparison to the model in Fig. 3.25.

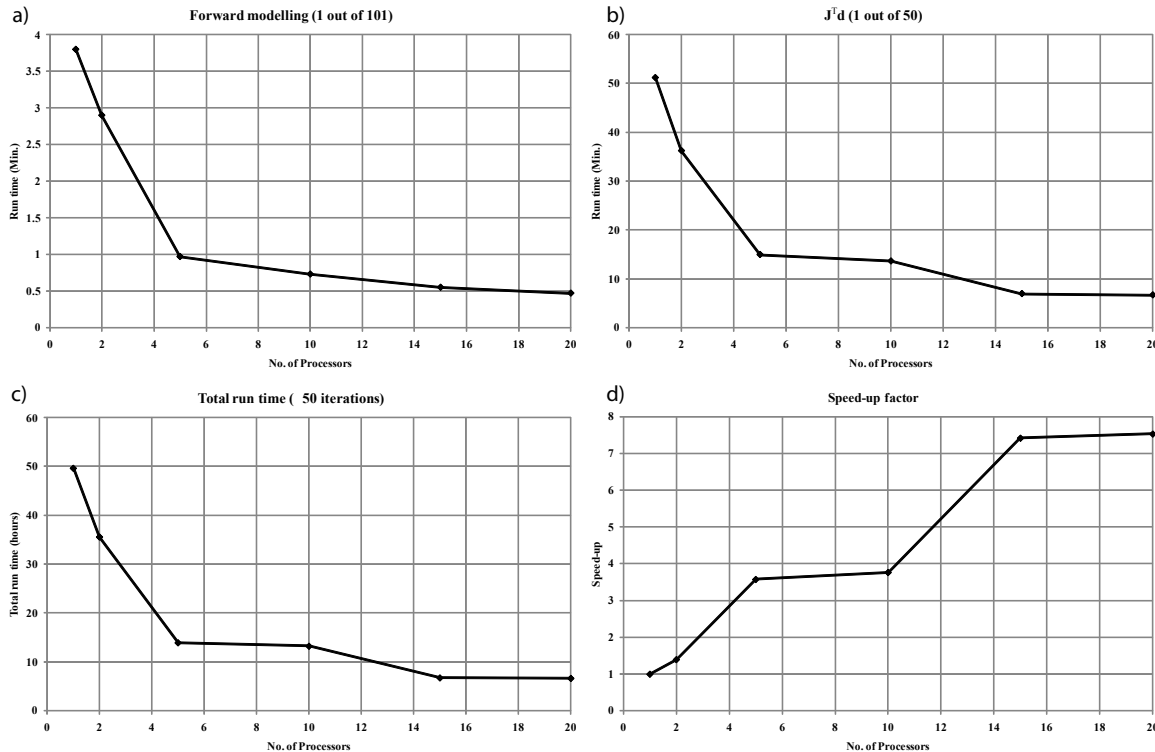


Figure 3.28: The run time curves for a) solving the forward modelling problem, b) computing the data gradients, c) the total run time to achieve 50 iterations and d) the speed-up factor as function of the used processors.

Figure 3.28 shows the run times for solving the forward modelling problem (a), computing the data gradients $\mathbf{J}^T \hat{\mathbf{d}}$ (b), the total run time for the 50 iterations (c) and the speed-up factor (d). The run time curve for solving the forward modelling problem decreases rapidly from 3.8 minutes to 0.97 minutes as the number of processors increases from 1 to 5. Afterwards, the run time decreases gradually to reach 0.5 minutes when using 20 processors. The run time curves for computing the data gradients (Fig. 3.28b) and the total run time (Fig. 3.28c) have the same behavior implying that the total run time is dominated by the run time to compute $\mathbf{J}^T \hat{\mathbf{d}}$. Figure 3.28c shows that the total run time to achieve 50 iterations is reduced from ~ 50 hours on a single machine to ~ 7 hours when using 20 processors in parallel which is equal to a speed up factor of 7.5 (Fig. 3.28d).

It is worthwhile to notice that the run times to solve the forward modeling problem can be further decreased if the parallelization runs over periods and modes. However, as pointed out above the total run time for the NLCG is dominated by the run time required to compute the data gradients. Therefore, parallelization over periods and modes to solve the forward modelling problem as part of the NLCG algorithm may not increase the overall speed up factor. However, parallelizing over periods and modes to achieve the computation $\mathbf{J}^T \hat{\mathbf{d}}$ is worthwhile.

In general, a comparison between the run time curves of the data space occam inversion (Fig. 3.26) and the NLCG (Fig. 3.28) reveals:

- The data space occam inversion requires less iterations due to its quadratic behaviour to converge to the desired error level. The quadratic behaviour, however, implies longer total run time because of the time consuming mathematical operations used in this algorithm.
- On the contrary, the NLCG algorithm requires more iterations to reach the global minimum, however, less total run time. This is because the total run time is dominated by the operation $\mathbf{J}^T \hat{\mathbf{d}}$ which requires less run time in comparison to computing the sensitivity matrix and achieving the multiplication $\mathbf{J}\mathbf{C}_m\mathbf{J}^T$.

Chapter 4

Magnetotelluric measurements in the southern Dead Sea basin

4.1 Introduction

As the largest pull-apart basin in the world, the Dead Sea basin (DSB) was and is an attractive geological structure for most geologists and geophysicists around the world. Understanding the geological and tectonic setting of the DSB may provide a better idea on how this geological structure formed and what the tectonic factors are that influence and control their formation and development.

Since the 1960ies, several geophysical measurements and geological surveys have been carried out to investigate the DSB. Among others, seismic, gravimetry and magnetics were the main geophysical methods that have been applied to study the deep internal and crustal structures of the DSB. Electrical and electromagnetic methods have been applied on both sides of the DSB, mainly to study the intrusion of saline water of the Dead Sea brines into surrounding fresh water aquifers. Up to date, there was no electrical or electromagnetic measurements along or cross the DSB to study its deep structures from a geoelectrical point of view.

The seismic data collected in the DSB area give a picture of the internal structure of the basin, however, along its main axis. This is due to the problematic political situation across the international boundary that runs parallel to the main axis of the DSB separating Jordan on the east from West Bank and Israel on the west. This political situation forced scientists from the three nations to run their own geophysical measurements in the Dead Sea area separately. A joint interpretation of the collected geophysical data from both sides of the boundary became possible in the 1990ies when scientists from the three nations started to exchange data and information. Nevertheless, there was no continuous traverse crossing the entire DSB in E-W direction to

study the faults pattern that control its western and eastern borders. A collaboration became only possible in the framework of a multi-national project called DESERT (Dead Sea Rift Transect). Under the leadership of the GeoForschungsZentrum (GFZ-Potsdam), scientists from Germany, Jordan, Palestine and Israel studied a segment of the Dead Sea transform (Wadi Araba) along a 300 km profile that extended from the Gaza strip in the north west to the Jordan highlands in the south east. Among others, magnetotellurics and seismics were the main geophysical methods that have been applied to study the deep structure of the Dead Sea transform at this segment. In the framework of the multi-disciplinary DESIRE (Dead Sea Integrated Research) project, several geophysical methods were applied along a profile to investigate the southern part of the DSB. Besides magnetotellurics, the methods used included active reflection and refraction seismic, aero-gravity, passive seismology, and thermo-mechanical modelling. The main propose of the DESIRE project is the development of an integrated geo-tectonic picture that can explain the development and the structure of DSB. Although the DSB is well studied in comparison to another basins along the DST, essential questions are still open.

With the magnetotelluric measurements across the DSB we intend to study the deep structures of the basin from a geoelectrical point of view and try to:

- Determine the location and depth extent of the major faults within the DSB.
- Image the thickness of the sediments within DSB.
- Image the internal structure of DSB.

4.2 Geological and tectonic background

The Dead Sea rift is a transform fault which extends from the Red Sea spreading centre in the south to the Taurus-Zagros continental collision zone in the north (Fig. 4.1a). The Dead Sea transform (DST) separates the Sinai plate in the west from the Arabian plate in the east with a total length of approximately 1000 km. The anticlockwise movement of the Arabian and the Sinai plates characterises the left lateral strike slip behaviour of the DST. The DST formed in the Cenozoic (65 million years ago) when the African-Arabian continent broke apart. Due to this movement, the Red Sea and the Gulf of Suez were formed (Garfunkel *et al.*, 1981). The offset of older geological formations, in particular of the late Cretaceous period (150 m.y. ago), reveals a total displacement between 100-110 km (Garfunkel *et al.*, 1981; Zak & Freund, 1981). Along the DST, Zak & Freund (1981) divided the shear movement of the DST into two main stages. An earlier shear movement of 60-65 km took place between 25 m.y. and 14 m.y. ago, whereas a later movement of about 40-45 km occurred during the last 4.5 m.y. The direction of shear movement along the Dead Sea was closer to the general trend of the rift (approximately N5°E) during the earlier

phase of the movement, and about 10° more to the east ($N15^\circ E$) during the later phase movement.

The average slip rate over the last 4-5 m.y. is estimated to be 7-10 mm/y (Garfunkel *et al.*, 1981). Based on geomorphological studies carried out recently in the northern part of Araba valley (a segment of the DST between the Dead Sea and the Gulf of Aqaba, Fig. 4.1a), Klinger *et al.* (2000) suggested an average slip rate of about 4 mm/y. Global Position System (GPS) measurements (Le Beon *et al.*, 2008) carried out in Araba valley estimated a slip rate of the DST of 4.9 ± 1.4 mm/y.

Faults are the dominant structural elements along the DST. Besides the predominate transform faults, normal faults are frequent along the margins of the rift valleys. Based on field studies, examination of topographic and geologic maps and study of aerial photographs, Garfunkel *et al.* (1981) delineated the active faults along a 500 km long segment of the DST from the Gulf of Aqaba/Elat to the Lebanon Mountains. The southern most segment of the DST, the so called Wadi Araba fault links the Red Sea basin in the south with Dead Sea basin in the north. The southern part of the Wadi Araba Valley is up to 10 km wide, and largely covered by Pleistocene and Holocene alluvium, playa deposits and some dunes. It is flanked by mountains in which the Precambrian basement and its sedimentary cover are exposed (Garfunkel *et al.*, 1981). The most prominent tectonic feature in this segment is the active left lateral strike-slip fault (Wadi Araba fault) which trends 19° - 20° to the east. Normal faults are also present in this area, typically dipping 60° - 80° to the east. Evidence for recent faulting is found on both sides of the Wadi Arabia main fault. Furthermore, the sharpness of the base of the eastern mountains indicates continuing uplift (Barjous & Mikbel, 1990; Garfunkel *et al.*, 1981). The elevation difference on both sides of the fault indicates that the uplift of the eastern rift shoulder was faster than on the western shoulder (Garfunkel *et al.*, 1981).

Going further to the north, the Wadi Araba valley starts to slope northward, draining into the Dead Sea. A change in the strike direction of the Wadi Araba fault is observed at the transition zone between the southern and northern parts of the fault. While the fault strikes between 19° - 20° in the southern part, it becomes 15° in the northern part. This change in the strike direction causes local compression and depression zones along the fault that can be expressed in terms of pressure ridges and rhomb-grabens (pull apart basin), respectively. The slip rate of the Araba fault in this area was estimated to be about 5-10 mm/y, by comparison of lateral displacements of three large alluvial fans on both sides of the fault (Garfunkel *et al.*, 1981).

To the north of the Araba valley begins the Dead Sea basin which consists of three distinct parts (Fig. 4.1b): the northern Araba valley, the southern basin of the Dead Sea and its northern basin (Garfunkel *et al.*, 1981; Garfunkel, 1981; Ben-Avraham *et al.*, 2008). The northern part of the Araba valley is covered by the Al-Lisan Formation. The transition to the southern basin of the Dead Sea is indicated by

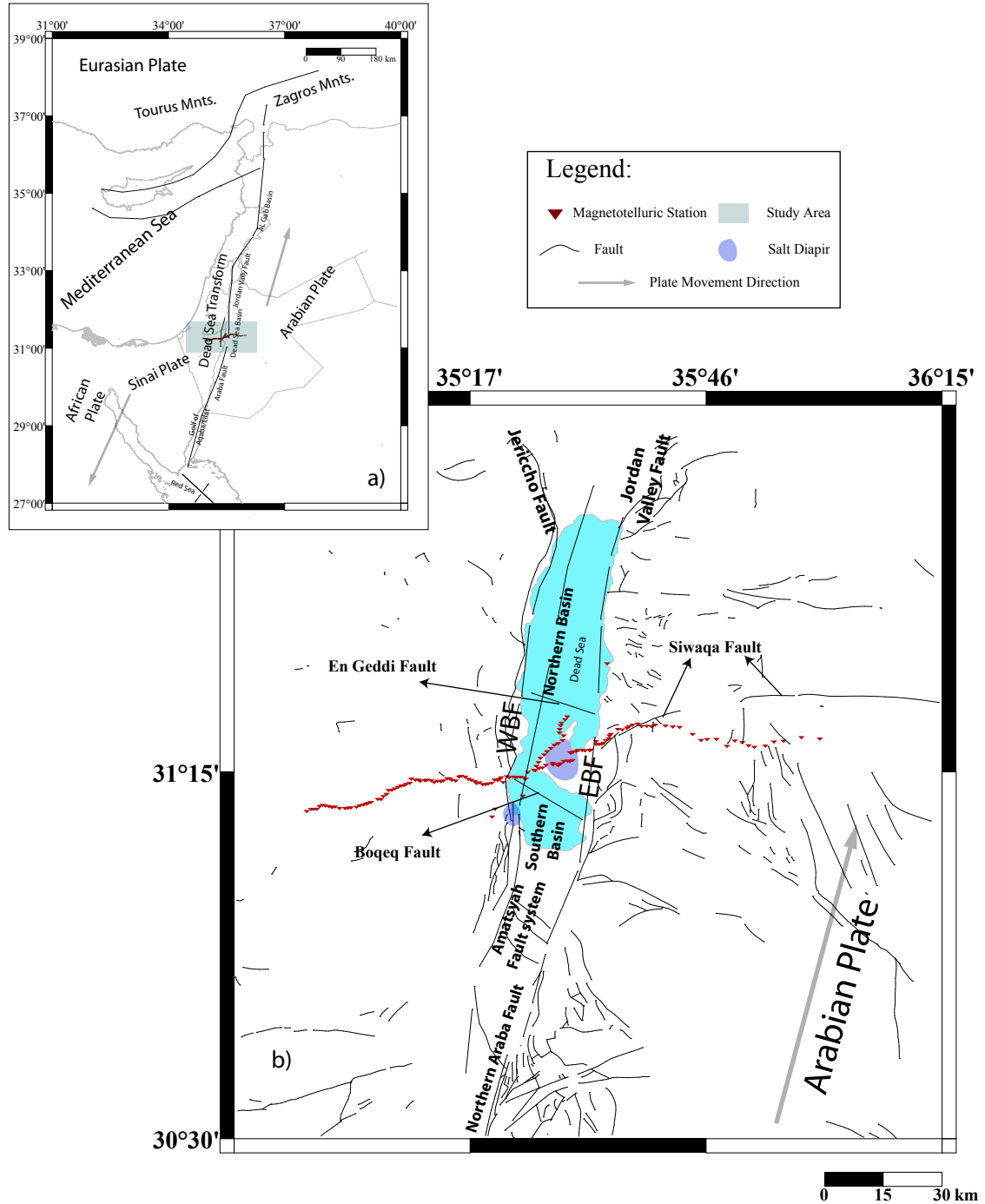


Figure 4.1: a) The major fault segments along the DST from the Gulf of Aqaba/Elat to Taurus-Zagros collision zone. b) Location map of the study area showing the position of the MT stations (red triangles) along the main and in-basin profiles together with the main faults in the vicinity of the DSB. The DST splits in the Dead Sea basin into the eastern (EBF) and the western boundary (WBF) faults, which delimit the DSB from east and west. Furthermore, the location and the extensions of the northern and southern sub-basins are shown.

the Amatsyahu normal fault system (Fig. 4.1b), which is NW-SE oriented (Garfunkel *et al.*, 1981; Al-Zoubi & Ten Brink, 2001; Ben-Avraham & Ten Brink, 1989; Hofstetter *et al.*, 2007). The Araba fault can be traced along the eastern margin of the valley. Further to the north of the Amatsyahu fault is the southern basin of the Dead Sea in which an extension of the Araba fault cannot be recognised at the surface. The southern basin of the Dead Sea extends from the Amatsyahu fault to the north of the Al-Lisan peninsula. The northern basin of the Dead Sea, which extends from the Al-Lisan peninsula to the northern part of the Dead Sea, is a rectangular shaped depression with a flat floor about -700 m below the sea level. Major faults extend along the borders of this deep basin. While the faults along the eastern side can be traced at surface, the western faults are covered by approximately 20-30 m of sediments (Garfunkel *et al.*, 1981).

Further to the north begins the Jordan valley which extends to the Sea of Galilee. Garfunkel *et al.* (1981) subdivided this segment of the DST into southern, central and northern parts. The southern and central parts of Jordan valley are covered mainly by the Al-Lisan formation. The Jordan Valley fault is the dominant fault in the area from the northwestern corner of the Dead Sea to south of the Sea of Galilee along the Jordan Valley (Fig. 4.1b). An indication of the left lateral strike slip along the Jordan Valley fault is the existence of the rhomb-graben, which are associated with irregularities of the fault in this segment of the DST (Garfunkel *et al.*, 1981). The northern part of Jordan Valley extends from the Sea of Galilee to the Mount Hermon in Lebanon.

4.2.1 The structure of the Dead Sea basin and its vicinity

The Dead Sea basin (DSB), is approximately 150 km long and extends from the southern Jordan valley near Jericho to the northern segment of the Araba valley (Fig. 4.1b). The area east of the basin formed a plateau (1.0-1.4 km above sea level) which descends gradually away from the transform. The area to the west formed a broad arch whose crest reaches only 0.6-0.8, rarely 1.0 km, above the sea level. The northern part of the basin is largely covered by the Dead Sea which is one of the most saline lakes in the world. Its waters contain > 30 % of dissolved salts, mainly Mg, Na and Ca chlorides (Neev & Hall, 1979). North and south of the lake the valley floor is 300-400 m below sea level, whereas about half of the lake floor is a flat rectangular area more than 700 m below sea level (water depth \sim 300 m). However, the southern sub-basin is sub-aerial with a maximum water depth of \sim 4 m (Ben-Avraham & Shubert, 2006). From a tectonic point of view, the DSB represents a large pull-apart basin, that has been formed along the Dead Sea Transform (DST) which contains a slight component of opening (Garfunkel *et al.*, 1981; Garfunkel, 1981): The DST splits into the northern segment of Araba fault on the southeast of the Dead Sea and the Jericho fault on the northwest of the Dead Sea. These two faults are arranged in en-echelon

(pull-apart) constructing the DSB (Garfunkel *et al.* , 1981). This means that the basin is delimited by fault scarps trending in N-S direction. However, the western fault scarp becomes indistinct along the northern Araba valley, so that the western border of the basin cannot be recognised.

The DSB is composed of two sub-basins separated by the Al-Lisan peninsula, the northern and the southern sub-basin (Garfunkel, 1981). Modelling of gravity data collected in the Dead Sea area has suggested that the southern part of the northern sub-basin has ~ 9 km of sedimentary fill (Ten Brink *et al.* , 1993). Furthermore, the model obtained from the gravity data suggests a similar basin structure for the southern sub-basin. The gravity data observed also a large negative anomaly over the Al-Lisan peninsula which has been interpreted that the basin has its maximum sedimentary fill (18 km thick) beneath the Al-Lisan peninsula (Ten Brink *et al.* , 1993). A recent study by Ben-Avraham & Shubert (2006) integrated with previous studies of Ginzburg & Ben-Avraham (1997), Al-Zoubi & Ben Avraham (2002), gravity data and borehole information suggests a slightly different structure concerning the sedimentary fill in the northern and southern sub-basins: The northern sub-basin seems to have a sedimentary fill of $\sim 6-8$ km. South of the Al-Lisan peninsula, a major fault running in SE-NW direction (the Boqeq fault, Fig. 4.1b) affecting the basement was detected. It downthrows the basement and the overlying sediments to the south by $\sim 4-5$ km. Thus, the Boqeq fault is considered to form the northern boundary of the southern sub-basin. Ben-Avraham & Shubert (2006) "The faulting was followed by the deposition of over 8 km of Pliocene to Recent sediments resulting in a 14 km thick sequence in the northern part of the southern sub-basin of the Dead Sea. The sedimentary basin in this area is therefore exceptionally deep with well defined boundary faults". Ben-Avraham & Shubert (2006) concluded that the deep southern sub-basin has the form of a "Drop Down" rather than a pull-apart basin. The integrated structures of the northern and southern sub-basins of the Dead Sea are plotted in Fig. 4.2.

The fault pattern in the DSB contains mainly of the left lateral strike-slip fault (Eastern boundary fault, EBF) which borders its eastern margin and the normal fault system denoted by WBF (Western boundary fault) which borders the western margin of the DSB (Neev & Hall, 1979; Garfunkel *et al.* , 1981). As pointed out above, the general trend of the faults along the DST and within the DSB is in N-S direction. Limited extension faults that runs in E-W direction can also be observed within the DSB (Ben-Avraham & Shubert, 2006). To the east of the Dead Sea, a major fault that runs in E-W directions can also be detected in the study area (the Swiaqa fault in Fig. 4.1b). The Swiaqa fault is a major wrench fault trending E-W at the easternmost end of the study area and rotating to the southwest near the Dead Sea (Khalil, 1992). The fault extends from the Dead Sea in the west to the Sirhan graben near the Saudi Arabia border (not shown in the map in Fig. 4.3). The fault is well documented and described by Khalil (1992) and later by Masri (2003).

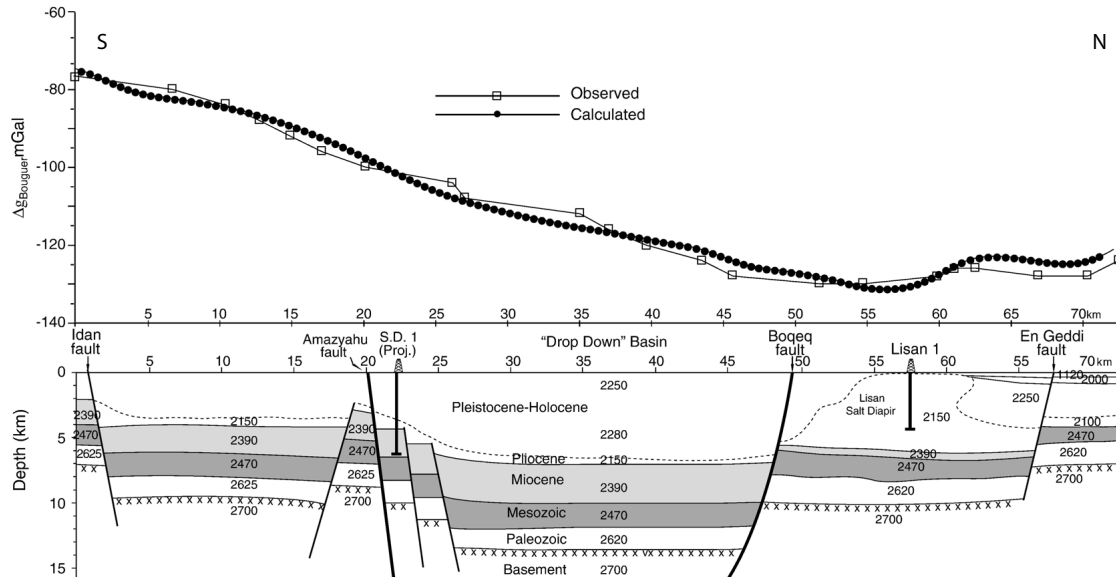


Figure 4.2: A north-south integrated geological cross section using seismic, gravity and boreholes data after Ben-Avraham & Shubert (2006). The cross section highlights the shape and the structure of the “Drop Down” which is delimited by normal faults: The Amazyahu fault in the south and the Boceq fault in the north. The Al-Lisan salt diapir under the Al-Lisan peninsula is also indicated in the cross section.

The most important characteristic of this fault, with respect to my work, is that the fault originated as weakness zone in the basement. Moreover, the fault is reactivated several times during different tectonic phases. An indication for the reactivation was a basalt intrusion (see Fig. 4.3) that is exposed at the western segment of the MT profile. This means that the fault must be older in age than the Pliocene. Another important point concerning the behaviour of the Swiaqa fault is that the fault has a northern downthrown in the study area with variable vertical displacement (Khalil, 1992).

The geological formations exposed in the study area vary in age from Quaternary to Precambrian (Fig. 4.3). However, Precambrian igneous rocks are exposed mainly to the south of the MT DESIRE profile. Along the MT profile, the geological formations vary in age from Quaternary to Cretaceous. Whereas, the most obvious geological formation that expose at the eastern segment of the MT profile is the olivine-basalt intrusion which extends for several kilometers along the MT profile. The Al-Lisan marl formation (alluvium, Quaternary) which is exposed over the Al-Lisan peninsula consists mainly of marl, clay-rich and clay-poor aragonitic marls (Powell, 1988; Khalil, 1992) with a maximum thickness of ~ 150 m. Further to the west, the MT profile extends mostly over the chalk, marl, clay, chert and limestone of the Senonian/Paleocene and less over the limestone of the Cenomanian/Turonian (upper Cretaceous) and the

alluvium of the Quaternary.

A recent study by Al-Zoubi & Ten Brink (2001) using seismic data from Jordan and

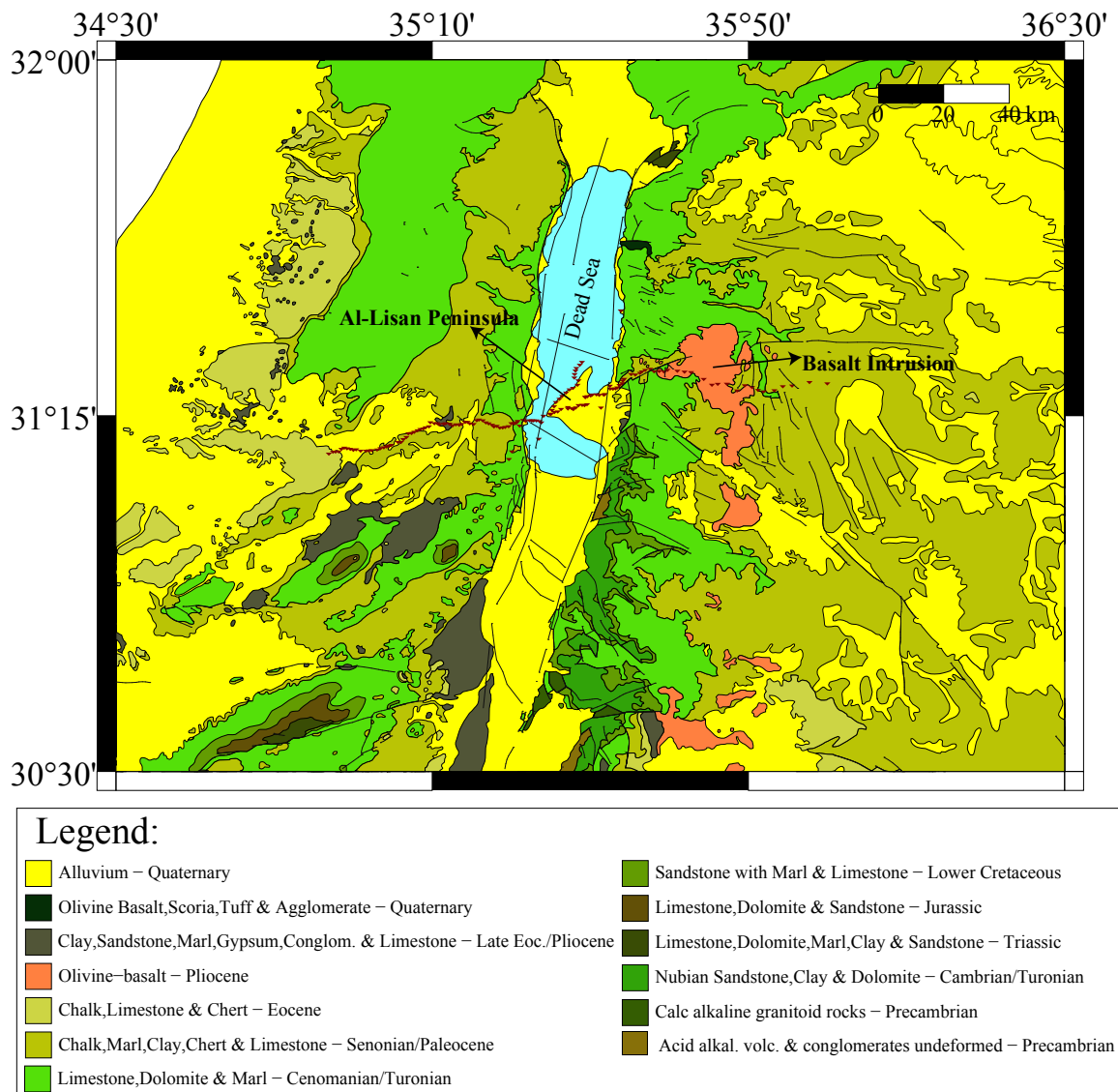


Figure 4.3: Map of surface geology showing the formations exposed in the study area. The MT profile (red triangles) runs mostly over sedimentary formations. Whereas, the igneous Precambrian rocks are exposed to the south of the MT profile. The basalt intrusion exposed at the eastern part of the profile is an indication of a recent activity along the Swiaqa fault.

Israel integrated with magnetic and gravity data focuses on salt diapirs within the DSB. Examples for those salt diapirs in the DSB are the large Al-Lisan diapir buried under the Al-Lisan peninsula and the small Sedom diapir which is exposed at the southwestern corner of the Dead Sea (Fig. 4.1). Based on models obtained from the collected geophysical data, Al-Zoubi & Ten Brink (2001) suggested that the Al-Lisan

salt diapir extends further to the south than previously assumed. However, the top of the salt body which is covered by a thin layer (200m-250m) of the Al-Lisan formation is deeper in the north than in the south. The authors also inferred that the Al-Lisan diapir reaches a maximum depth of 7.25 km. On the other hand, the Sedom diapir in the south seems to have less depth extend (5.5-6 km). The results presented by [Al-Zoubi & Ten Brink \(2001\)](#) show an interesting shape of the Al-Lisan salt diapir: The contour map used to plot the depth to the top of the salt diapir (Fig. 4.4) shows that the depth to the top increases rapidly at the eastern border of the diapir and gradually at the western border. This gives an impression that the eastern top of the Al-Lisan diapir is almost vertical and the western surface is more flat. The contour map in Fig. 4.4 also shows a deep evaporitic layer in form of a syncline between the Al-Lisan and the Sedom salt diapir. This evaporitic layer is interpreted to be the source salt layer that was deposited during the Pliocene directly after formation of the Dead Sea basin in the Miocene with a total thickness of ~ 2.3 km ([Al-Zoubi & Ten Brink, 2001](#)). Accumulation of ~ 4 km sediments in the Dead Sea basin during the Pleistocene may have cause a static pressure that forced the evaporitic layer to uplift in form of diapirs.

4.3 Magnetotelluric field work and data processing

The MT profile (Fig. 4.5) follows roughly the seismic profile ([Mechie *et al.*, 2009](#)), crossing the Dead Sea Transform (DST) at a place where the fault separates into an eastern and western border fault (EBF and WBF), where the basin fill is supposed to be at its largest thickness (~ 10 km; [Ben-Avraham & Grasso \(1991\)](#)) and where the intruded Al-Lisan salt diapir formed a three-dimensional feature.

The MT data were collected in October and November 2006. The measurements were carried out along two profiles. The main profile is oriented N70°E, approximately perpendicular to the surface trace of the DST and has a length of ~ 110 km. A second, shorter profile (~ 20 km) is oriented N20°E and runs parallel to the west coast of the Al-Lisan peninsula (Fig. 4.5).

In total, we have deployed 151 MT stations distributed along the two profiles. The distance between the stations varies between 0.5 and 2 km depending on the location: In the Dead Sea basin and its vicinity, 0.5 km was the distance between two stations; away from the DSB and at the eastern segment of the profile, the distances between the stations increased up to 2 km in the outer parts.

The 151 MT stations were deployed using mainly two data recording systems: Short Period Automatic Magnetotelluric (S.P.A.M. MkIII) and Earth Data Logger (EDL) instruments. The S.P.A.M. MkIII is a real-time system. It was used in the field for two main proposes: To insure that the site-setup is connected correctly (coils, sensor boxes, cables, etc) and to run the measurements in the short period range (2 kHz to

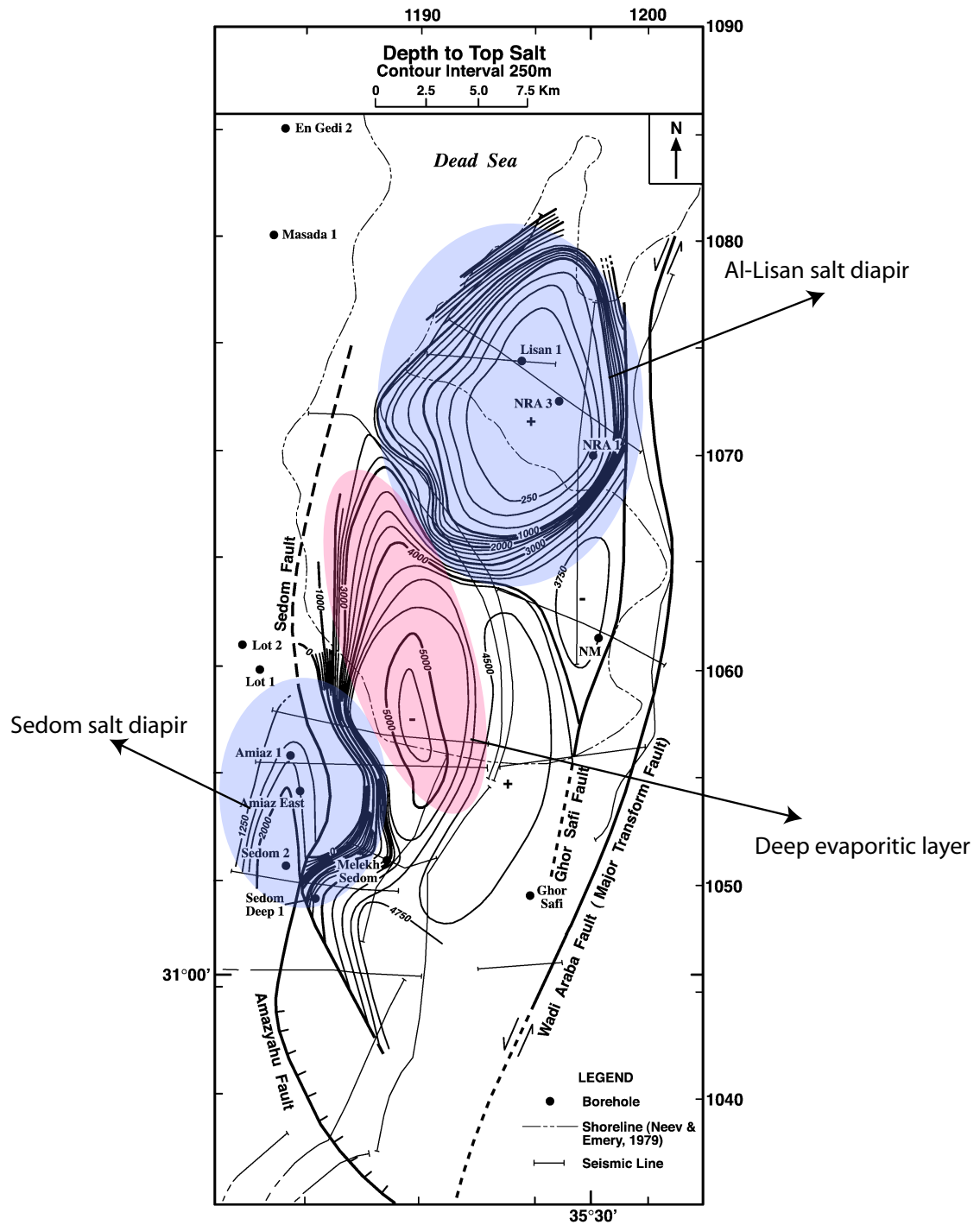


Figure 4.4: A map showing the depth to the top of salt in the Dead Sea basin modified after Al-Zoubi & Ten Brink (2001). The contour interval is 250m. The depth to the top of the Al-Lisan salt diapir increases rapidly at the eastern border and gradually at the western border. The same behavior can be observed at the eastern border of the Sedom salt diapir. The deep evaporitic layer between the Al-Lisan and the Sedom diapirs was interpreted by Al-Zoubi & Ten Brink (2001) as the source of the salt.

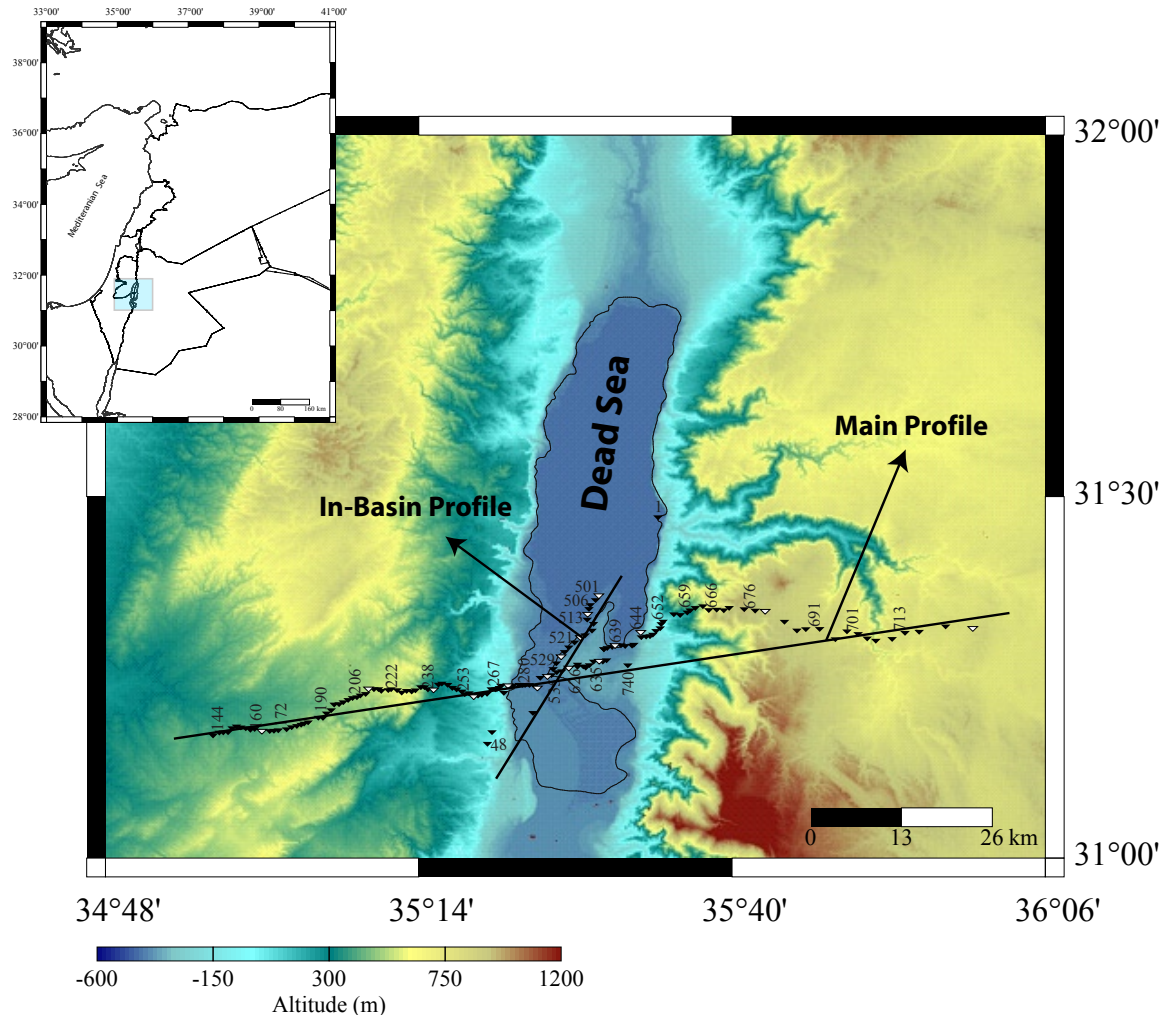


Figure 4.5: Topographic map with the location of the MT stations along the main and the in-basin profiles. Black triangles indicate the locations of the MT broad band (BB) stations, whereas white triangles represent the locations of combined BB/LMT stations.

$\sim 10\text{Hz}$). For the latter, the total run time at each station was 24h. At all stations, induction coil magnetometers were used for the broad band (BB) stations to record the time variations of the three orthogonal magnetic components (B_x , B_y and B_z). The EDL system was used for the BB stations with two sampling rates (50Hz and 500Hz) to cover the period range between 0.004s-1000s. The total recording time for each BB stations was three days. In addition to the BB stations, 17 Long Periods MT stations (LMT, white triangles in Fig. 4.5) covering periods longer than 1000s were also deployed and distributed along the two profiles. However, for the LMT stations, flux-gate magnetometers were used with a sampling rate of 2s. The total run time for each LMT station was approximately one month.

Data acquisition was accomplished with two independent teams, working at the same

time in Jordan and Israel which allowed to operate up to 30 sites recording simultaneously. This provides great flexibility to use remote reference processing with many combinations of sites.

Besides a main remote reference station (number 501; see Fig. 4.5) which recorded time series for 5 weeks, many normal BB sites with a good data quality could be used as remote reference stations, presuming an overlapping in the recording time. Figure 4.6 presents a part of the stations run times for the time period from 31 October to 4 November 2006. The blue bars in Fig. 4.6 denote stations located to the east of the Dead Sea in Jordan. The red bars indicate stations located to the west of the Dead Sea in Israel. As can be seen in Fig. 4.6, there are several stations from both sides that have overlapping recording time.

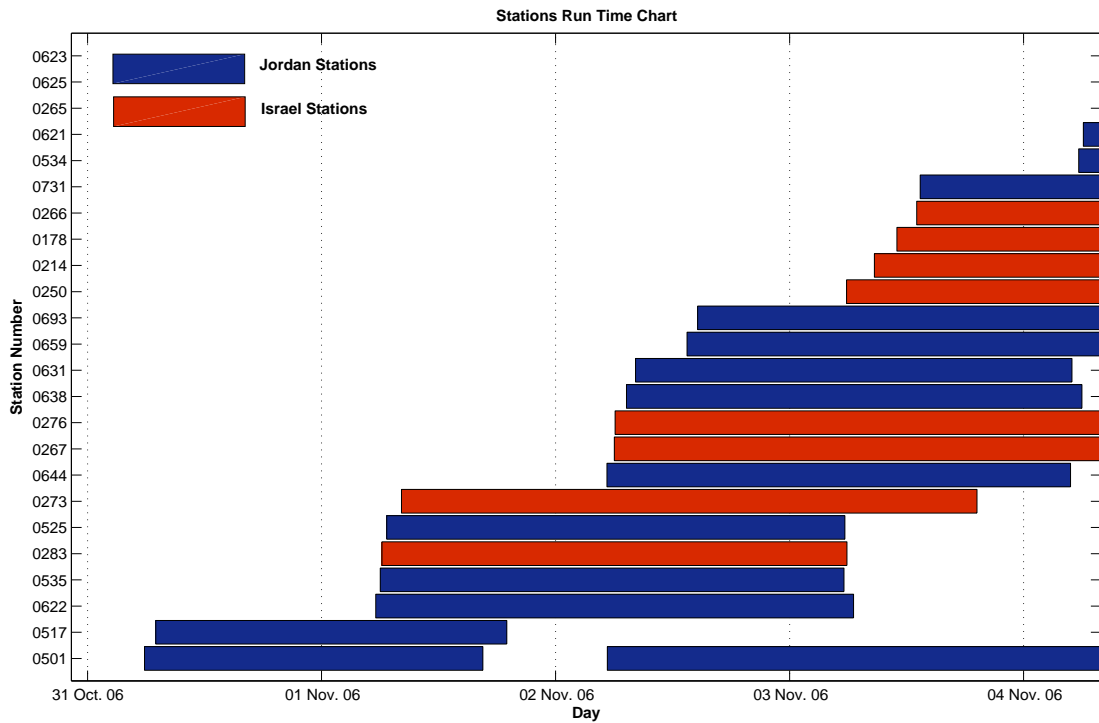


Figure 4.6: Part of station run time chart for stations from Jordan (blue bars) and Israel (red bars) for the time period between 29 October and 4 November 2006. This chart is used for the remote reference processing to identify overlapping recording times between different stations.

The recorded time series were processed using the EMERALD package (Ritter *et al.*, 1998; Weckmann *et al.*, 2005; Krings, 2007) to estimate the impedance tensor and the vertical magnetic transfer functions.

Figures 4.7a and 4.8a show apparent resistivities, phase curves of both off diagonal impedance tensor elements, and induction vectors for two stations from different segments of the profile processed by applying single site (SS) processing technique. As

can be seen in i.e. Fig. 4.7a, the data are affected by noise in the period range around 10s (dead band). On the other hand, data in the long period and high frequency ranges are less affected by noise. To improve the data quality, remote reference (RR) processing was applied. This method is based on using the horizontal magnetic field components of the local site together with the magnetic field components from a remote site. If noise between the local and reference sites is uncorrelated, the bias in the functions can be eliminated (Gamble *et al.*, 1979; Egbert, 1997).

To demonstrate the importance of the remote reference processing in improving the data quality, two broad band stations (3 days recording time, covering the period range 0.001s- 1000s) from different segments of the profile are discussed as an example (station numbers 202 and 529, see Fig. 4.5 for location).

For the RR processing, stations number 192, 501 and 210 were used as remote reference stations to improve the data quality of station number 202. The recording time of station number 192 is partially overlapping with the recording time of station 202 (1 day). Figure 4.7b shows some improvement of the data quality of station 202 in the dead band range. However, due to the short overlapping recording time of only one day, data in the long period range become worse. Moreover, the induction vectors from 10s to the longest period are more scattered in comparison to the induction vectors obtained using SS processing. The recording time of station number 210 completely overlaps with the recording time of station 202. Figure 4.7d shows that the use of station 210 improves the apparent resistivity values ρ_{yx} of station number 202. However, the phases from the same mode (ϕ_{yx}) at longer periods ($> 100s$) are not improved.

Figure 4.7c presents the result of the RR processing using station number 501 as remote reference station which had been recording for 5 weeks and was chosen because the data seem to be relatively free of noise. Using station 501 as a remote reference station improves significantly the data quality of station 202 in the dead band range and also improves the quality of the long period data. Figure 4.8 shows a similar procedure used to process station 529 using SS and RR as used for station 202.

The results obtained from the RR processing can be summarised as follows:

- All remote reference stations can improved the data quality up to 1s (Figs. 4.8b, 4.8c, and 4.8d).
- Only the remote reference station number 501 was be able to improve also the data quality at longer period (up to 100s; Fig. 4.8c).
- For periods $> 100s$, none of the remote reference stations could improve or at least reveal the data quality obtained by SS processing.

Since the RR processing does not always result in better data quality than SS processing, a combination of the results from RR and SS processing was used. This means,

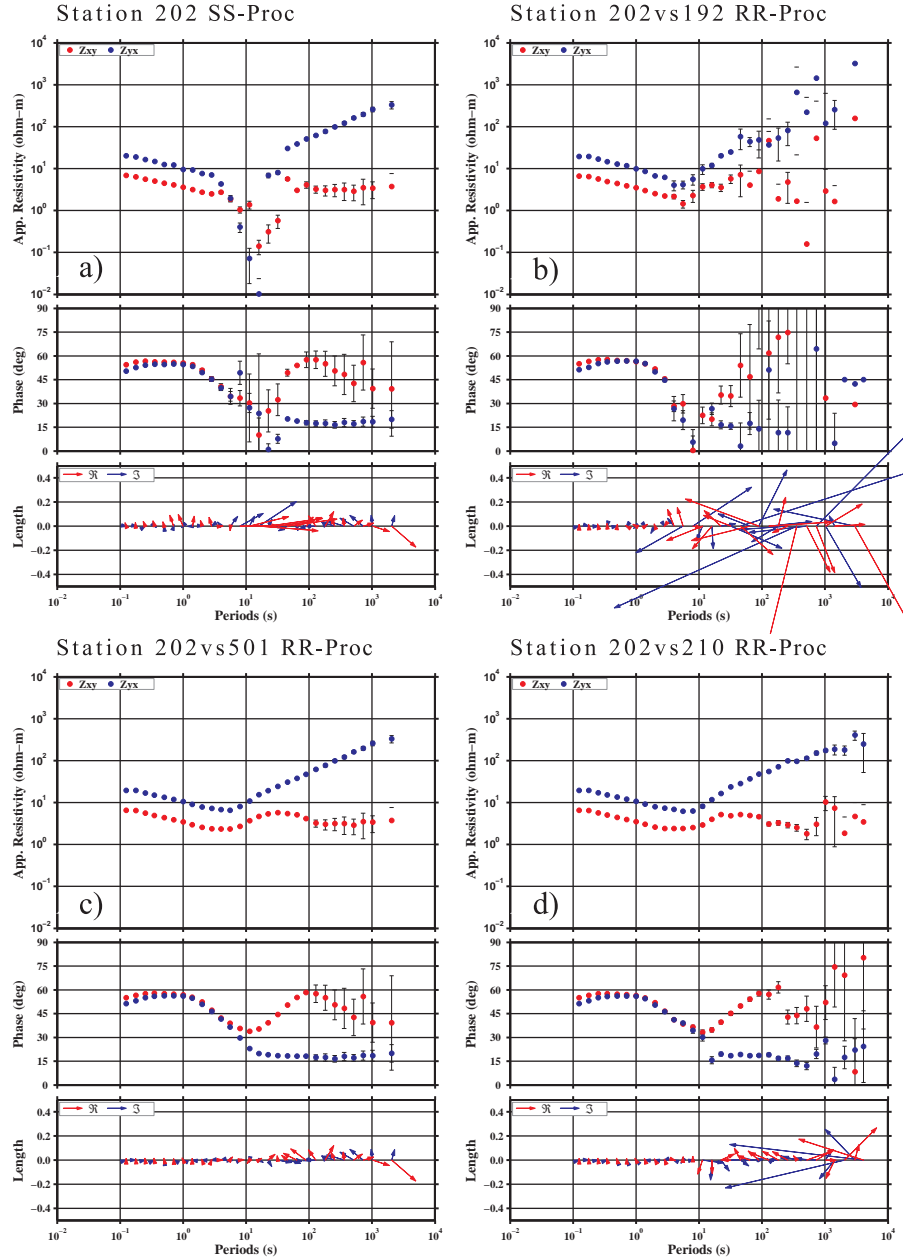


Figure 4.7: Comparison between single site (SS) and remote reference (RR) processing for station number 202. a) Results for a SS processing shown as curves of apparent resistivity (Z_{xy} , red points; Z_{yx} , blue points; upper panel), phase (middle panel), and induction vectors (Real part, red arrows; imaginary part, blue arrows; lower panel); all parameters are functions of period. All three quantities (app. resistivity, phase and induction vectors) are strongly affected by noise in the period range 5s-80s. b) RR processing using station number 192 as remote reference for station number 202 improves the data in the affected period band. However, data at longer periods ($> 100s$) became worse. c) Using station 501 as remote reference improves the data quality for station 202 and keeps the data quality at longer periods ($> 100s$) unaffected. d) Using station 210 that recorded the data simultaneous with 202 as remote reference improves the data quality in the affected period band. However, the data at long period are still bad in comparison with the SS result

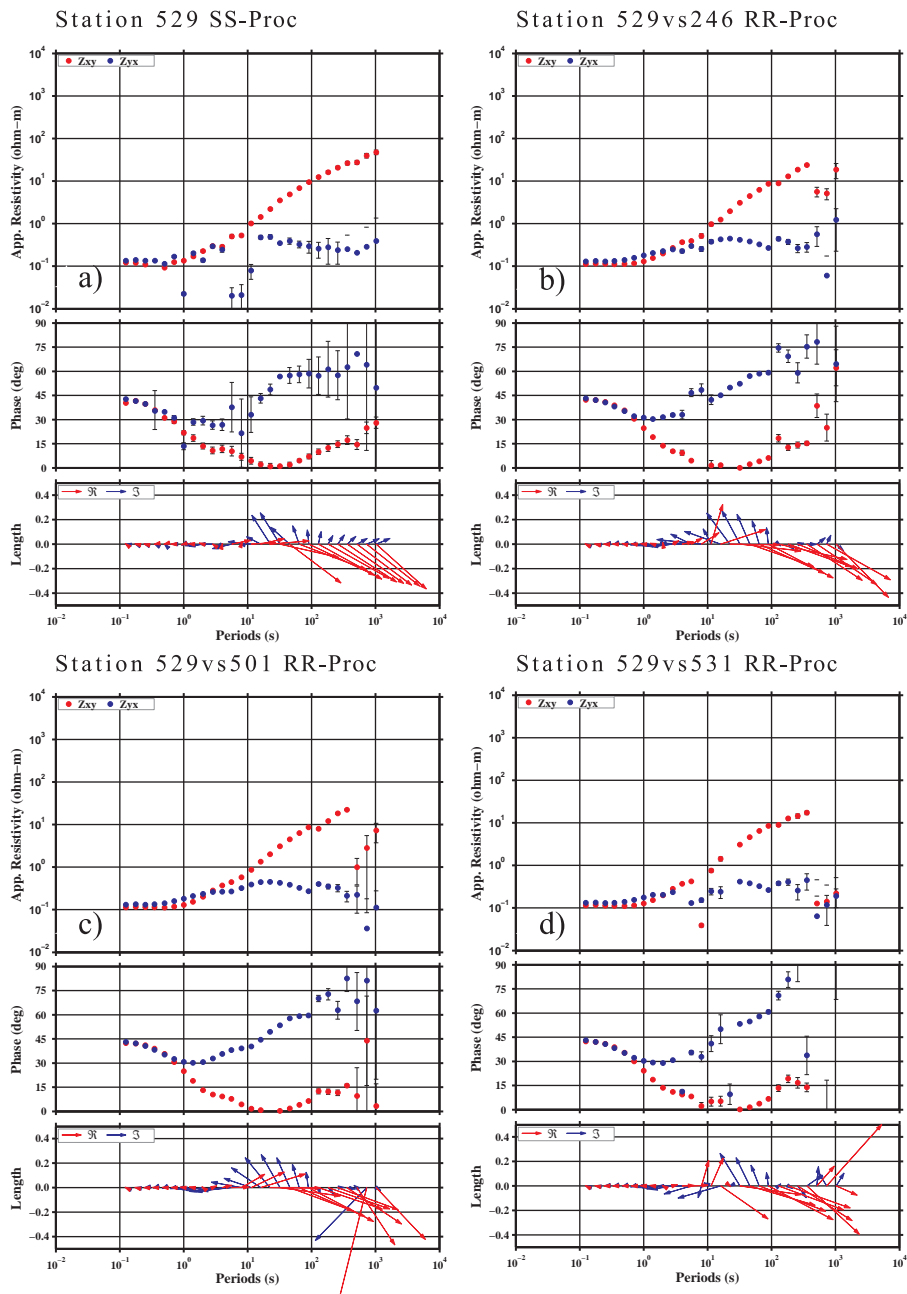


Figure 4.8: Data example of station 529 to demonstrate the effect of the RR processing. a) Results from SS processing: The noisy period band at this station extends from 0.5s-10s. b) Using station 246 as a reference for station 529 improves the data quality in the short period range (up to 5s). A slight improvement is achieved for periods between 5s and 10s. c) A significant improvement in the data quality of the affected period band can be achieved when using station 501 as remote reference station. However, none of the used RR stations was able to improve, or at least to return the same data quality as the SS processing for the long period data. d) Using station 531 as reference does not improve the data quality in the affected period band.

the final result for each station may contain data obtained from SS or RR processing.

4.4 Regional geo-electric strike direction and dimensionality analyses

After processing, the data are analysed to estimate the regional geo-electric strike direction and to investigate their dimensionality. For a 2D interpretation, it is essential to rotate the data into the geo-electric strike direction to set the measurement axis parallel (E-polarisation) and normal (B-polarisation) to the geo-electric strike direction.

To have an independent idea about the conductivity distribution, in particular vertical discontinuities, and to analyse the regional geo-electric strike direction in the study area, the induction vectors are first discussed. Induction vectors are complex quantities representing the ratio of vertical to horizontal magnetic fields components (eqs. 2.33 and 2.35). Since a vertical magnetic field component occurs only in the vicinity of horizontal electrical conductivity gradients, the induction vectors can be used to indicate the presence or absence of lateral variations in conductivity (Weaver, 1994). In the Wiese convention (Wiese, 1962), the real part of the induction vectors (eq. 2.35) points away from a conductor where the electric currents have their maximum concentration (Weaver, 1994).

Figure 4.9 shows the real part of the induction vectors plotted for the DESIRE-MT data at different periods¹.

At almost all stations, the length of the induction vectors at shorter periods ($< 1s$) is almost zero, except for a few stations at the westernmost part of the profile. At these stations the induction vectors are affected by very local features (or noise). From 4s to 1024s the amplitudes of the induction vectors at stations located at the central and eastern parts of the profile start to increase gradually. The behaviour of the induction vectors in the central part of the profile, can be divided into three main groups:

- Stations located close to the western coast of the Dead Sea, where the induction vectors point to the west. The amplitude for these stations starts to increase from 4s up to 256s and decrease at longer periods ($> 256s$).
- Stations located to the east of the Dead Sea and along the western coast of Al-Lisan peninsula, where the general direction of the induction vectors is 135° . Few stations located directly at the central part of the Dead Sea basin show a direction of induction vectors to the east. The amplitudes of the induction

¹Outliers are not shown.

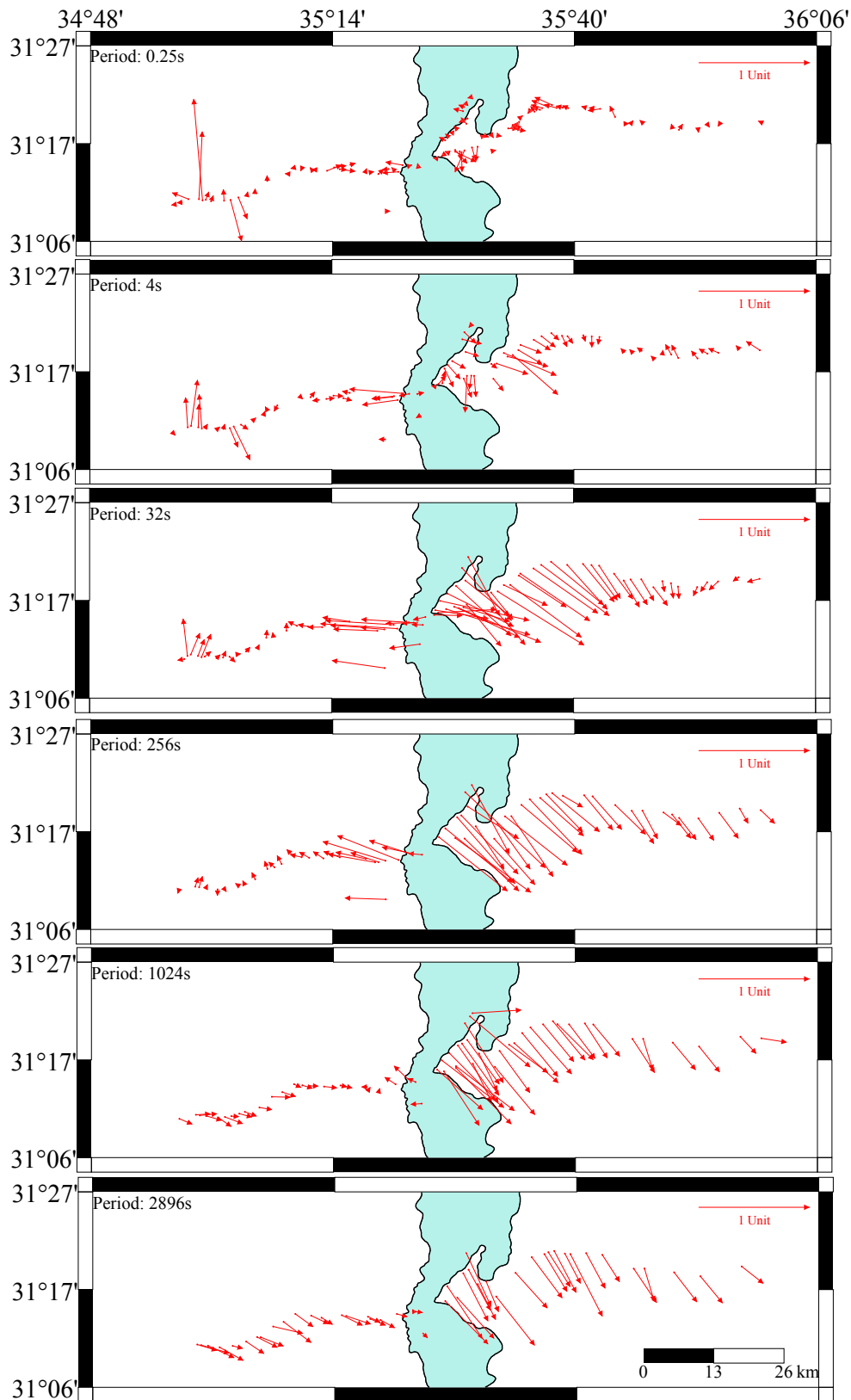


Figure 4.9: Real part of the computed induction vectors at all stations (main and in-basin profiles) for different periods. See text for explanation.

vectors at all of these station starts to increase at 4s, showing high values even at the longest periods.

- At periods > 256 s the amplitude of the induction vectors at the westernmost stations start to increase gradually; they point to the west with a slight rotation to the south at the longest period (2896s).

The directions and the amplitudes of the induction vectors in the central part of the study area are an indication of a conductor with N-S orientation which starts at relatively shallow depth and continues towards greater depth. In addition, a considerable effect from the very conductive Dead Sea brines and the underlying sediments in the north can be noticed in the induction vectors. Furthermore, the southward rotation of the induction vectors at periods between 32s-2896s at the eastern stations indicates that a conductive body is located to the north of these stations. At periods longer than 256s, the directions of the induction vectors at the westernmost stations indicate the existence of a conductive structure that is located to the west of the study area with an approximately N-S orientation. The depth of this conductor could be either shallow, however, laterally displaced to the west of the study area or located relatively deeply close to the western end of the main profile. The first case is more plausible, as the conductive Mediterranean Sea and its underlying sedimentary basins are located approximately 100 km to the west of most western station. Moreover, the induction vectors could image the ocean effect.

This data analysis shows that the electrical conductivity distribution in the study area is strongly correlated with the orientation of the Dead Sea and its basin architecture. Therefore, the first approximation of the regional geo-electric strike direction is N-S. A quantitative determination of the regional geo-electric strike direction can be derived from the impedance tensor (\mathbf{Z}), with so-called tensor decomposition methods. Two methods are widely used in determining the geo-electric strike direction: The method proposed by [Bahr \(1988\)](#) and the method described by [Becken & Burkhardt \(2004\)](#).

The method proposed by [Bahr \(1988\)](#) is based on decomposing the impedance tensor into a non-inductive and inductive response, taking into account that the phases between the columns of the tensor are equal. Therefore, this method is usually referred as a phase-sensitive strike.

Figure 4.10a presents the estimated strike direction using the Bahr method as a function of periods and using all stations. Figure 4.10a illustrates that for periods > 1 s the majority of stations show a strike direction of 350° changing slightly towards 10° for longer periods. Please note that in Fig. 4.10a only one quadrant is displayed and that the values from other quadrants are mirrored into the first one. To analyse the strike direction along the profile, the same result as in figure 4.10a is used, however, as a function of the station position. Figure 4.10b shows that the station

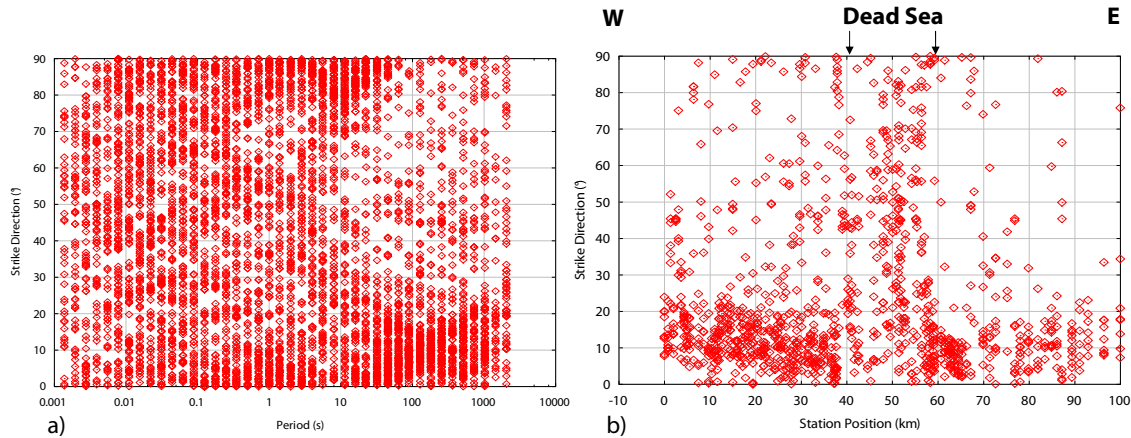


Figure 4.10: Phase-sensitive geo-electric strike directions computed using Bahr's method presented as a function of a) periods and b) station position in the period range 50s-10000s. a) Indicates that the strike directions change from 350° for periods $< 10s$ towards 10° for longer periods ($> 10s$). The result presented in b) shows obviously that at the western and eastern stations the strike directions cluster around 10° . Whereas, at the central stations there is no preferred strike direction. Please note that four quadrants are displayed in one.

can be clustered in three groups. Stations located to the east and to the west of the Dead Sea indicate a strike direction of approximately 10° in the period range from 50s to the longest period. Stations located in the Dead Sea area do not indicate any pronounced strike direction.

The second method that I applied to the measured data to estimate the regional geo-electric strike direction is the method described by [Becken & Burkhardt \(2004\)](#). This method is based on an ellipse parameterization of the impedance tensor columns. The ellipticities of the impedance tensor columns are rotationally variant parameters. They vanish for 2D electromagnetic field conditions if the data are rotated to the regional strike direction. Therefore, estimating the regional strike direction is achieved by minimising the sum of squared ellipticities weighted with their variances by rotating the coordinate system. The minimisation procedure can be applied for single periods and sites as well as for a range of periods and/or sites simultaneously, which improves the stability and accuracy of the estimated strike direction.

Based on the results shown in Fig. 4.10b, I divided the stations into three subsets: stations located to the east and to the west of the Dead Sea and stations located in the Dead Sea area (Fig. 4.11). The regional strike direction is computed using the method of [Becken & Burkhardt \(2004\)](#) for each subset separately using only data from long period range 1s-1000s (Fig. 4.11a, b and c).

The strike direction for sites located to the west and to the east of the Dead Sea (purple and green profile segments, respectively, in Fig. 4.11) exhibit predominantly

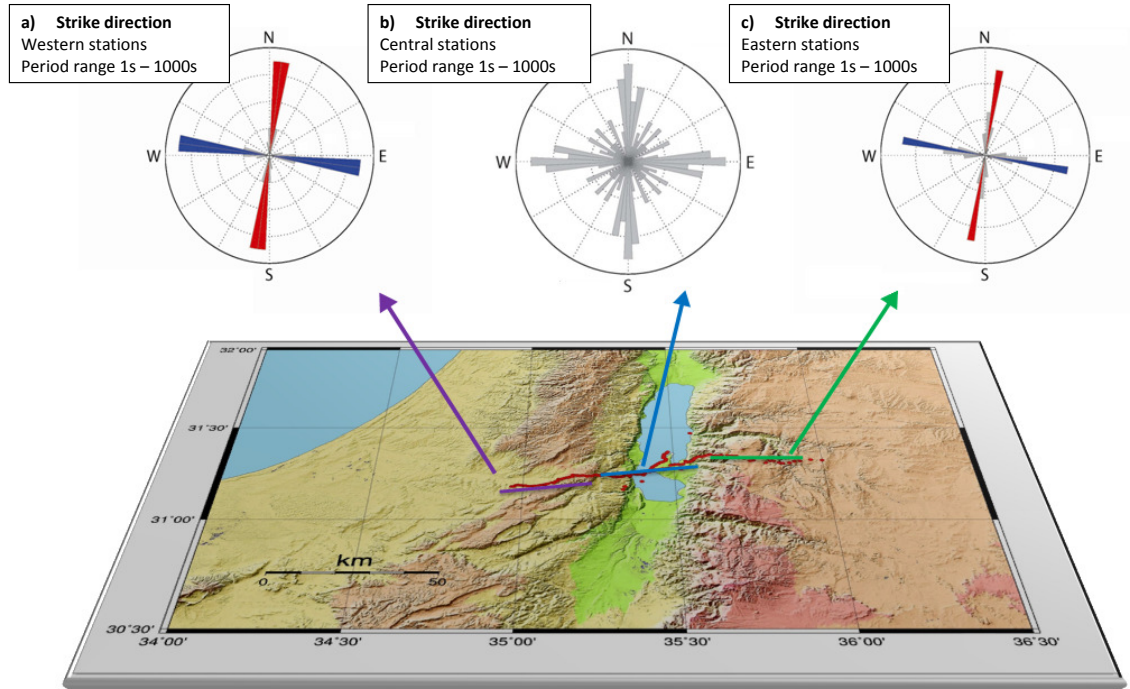


Figure 4.11: The computed strike direction using the method described by Becken and Burkhardt (2004) in the period range 1s-1000s at different segments of the main profile. The rose diagram of the strike direction at the western and eastern segments (a and c) indicate a dominant N-S strike direction. Whereas, the strike direction at central stations (b) varies between 0° - 90° . However, one can still recognise a N-S strike direction. The blue sectors in the rose diagram a and c show the 90° ambiguity in the calculated strike direction.

2D conditions with a very stable strike direction of 10° - 15° E (90° ambiguity; cf. rose diagrams in Fig. 4.11a and c, respectively). In the central segment of the profile, the geo-electric strike direction exhibits greater variability (see rose diagram in Fig. 4.11b) and reveals a more complicated, three-dimensional (or one dimensional) nature of the internal structure of the Dead Sea area.

The behavior of the geo-electric strike direction along the main profile determined by the two methods presented in Figs. 4.10 and 4.11 can be summarised as follow:

- Stations located to the east and to the west of the Dead Sea reveal a stable strike direction between 10° - 20° at periods >10 s. This is an indication that a 2D structures dominate these two segments of the main profile.
- Stations located in the Dead Sea area show strong variation in the strike direction values. This reflects the complexity of the geological and tectonic setting of the Dead Sea basin. Nevertheless, for these stations and in the period range 1s-1000s one can still recognise a predominate strike direction in N-S (Fig. 4.11b).

A parameter to analyse the dimensionality of sub-surface structures and the associated electromagnetic fields is typically called *skew*. In the original approach, the skew was calculated by the magnitude of the complex ratio between the main diagonal and the off-diagonal elements of the impedance tensor. In case of ideal 2D structures, the value of the skew is zero, since $Z_{xx} = Z_{yy} = 0$. However, in 3D structures the value of the skew may reach 1 and above, depending how complex is the electromagnetic field above the structures. Bahr (1991) suggested an empirical threshold value of 0.3 above which the dimensionality of the electromagnetic field can be considered 3D. Similar to the determination of the strike direction, the skew is also computed in a phase-sensitive way using Bahr method (Bahr, 1988). Figure 4.12a and b present the results of the phase-sensitive skew as a function of periods and station position, respectively. Figure 4.12a shows that up to 10s, most of stations have skew values less than 0.3. At periods >10s, the skew values increase for the majority of stations.

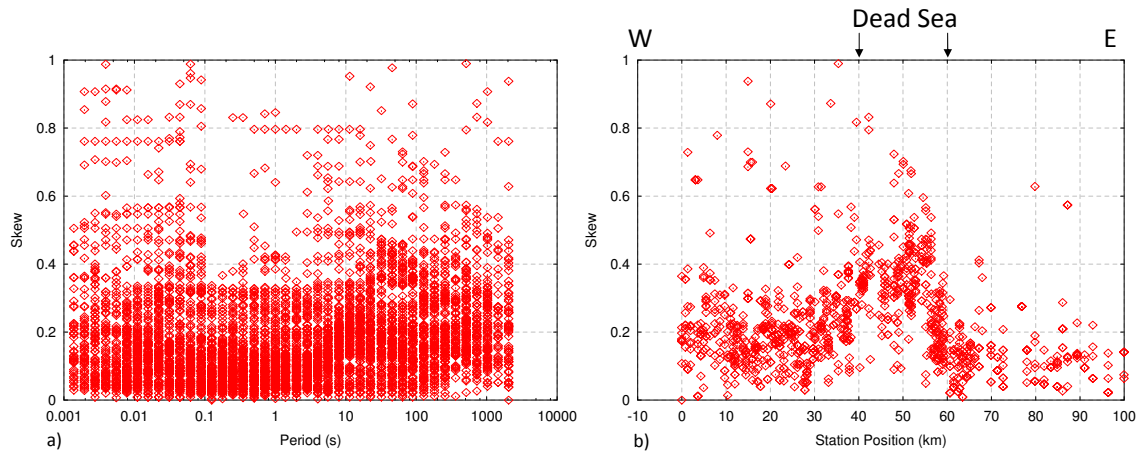


Figure 4.12: The computed skew values using Bahr’s method as a function of a) periods and b) station position. The gradually increase of skew values at periods longer than 10s reflects the 3D nature of the data. The results presented in b) indicate that the data from the western and eastern stations behave more 2D than 3D (skew < 0.3). However, the data from stations located in the Dead Sea area have a higher value of skew (> 0.3) which reflect 3D structures in this area.

Figure 4.12b presents the results of the phase-sensitive skew as a function of station position. Now one can distinguish between three segments of the profile that behave differently concerning the computed skew values. Most of stations located to the west and to the east of the Dead Sea have skew values ≤ 0.2 . However, in the central part of the profile, the skew values are significantly higher than 0.3.

The regional geo-electric strike direction estimation (Fig. 4.11) and the dimensionality analyses using the phase-sensitive skew values (Fig. 4.12) can be summarised as follows:

- According to the obtained strike directions and skew values, the stations are clustered into three groups: East, west and in the Dead Sea basin.
- The computed strike directions and skew values for both eastern and western stations give an indication that the regional structures below these two segments of the main profile are 2D with a 10° - 15° regional geo-electric strike direction.
- Stations in the central part of the profile do not show clear strike directions and higher skew values. This indicates that the structures below this segment of the profile have a 3D nature.

4.5 Interpretation of the DESIRE magnetotelluric data

4.5.1 Evaluation of the measured data

After rotating the impedance tensor into the geoelectrical regional strike direction of 12° (see previous section) the data are assigned to E- and B-polarization. For the data in the E-polarisation mode (TE-mode), the electric currents flow parallel to the strike direction. Whereas, the electric currents that flow perpendicular to the strike direction are assigned to the B-polarisation data (TM-mode).

A common way to present multi-site, multi-period data sets is with so-called pseudo-sections of apparent resistivities and phases for both TE- and TM-modes. In the pseudo-sections, the apparent resistivities and phases are usually plotted as a function of profile distances (X-axis) and as a function of period (Y-axis). The commonly used colour convention in the electromagnetic community is to assign the red and yellow colours for low and blue colours for high values of the apparent resistivities. Since low values of the impedance phases are correlated with high values of the apparent resistivity the opposite colour convention is used for the phase values.

Figure 4.13 shows the apparent resistivities and phases as pseudo-sections for TE- and TM-modes for all stations from the main profile of the DESIRE-MT data set.

The most obvious features in the apparent resistivity pseudo-sections in Fig. 4.13 are very low apparent resistivity values ($< 1 \Omega m$) which are observed at stations in the Dead Sea area (label A in Fig. 4.13). In the TE-mode, this feature covers the period range from 0.001s to 50s. Low TM-mode apparent resistivity values, however, can be observed to about 1000s, particularly at stations in the westernmost part of the Dead Sea area. The corresponding phase values of this feature reach 70° in the TM-mode. The apparent resistivities values for the TE-mode at longer periods ($> 10s$) do not indicate the same behaviour as in the TM-mode (label B in Fig. 4.13). Instead, the apparent resistivity values vary between 10 and $100 \Omega m$. The TE-mode phases at longer period ($> 10s$) show values greater than 45° . This discrepancy between TM- and TE-modes at periods longer than 10s in both apparent resistivities and phases values indicates the existence of at least 2D structures at depth.

Stations located in the Dead Sea area indicate another interesting feature that can be observed in the apparent resistivities and phases in both TE- and TM-modes (label C). At periods between 1s and 10s, an increase in the apparent resistivity values is observable. The corresponding phases show low values of 10° for the TM-mode and around 20° for the TE-mode. This relatively small feature in the data set indicates a resistive structure embedded in a highly conductive host.

At periods shorter than 10s, data from stations located to the east and to the west of the Dead Sea indicate a similar behaviour in both modes (D), this indicates that the shallow structures may have a 1D nature.

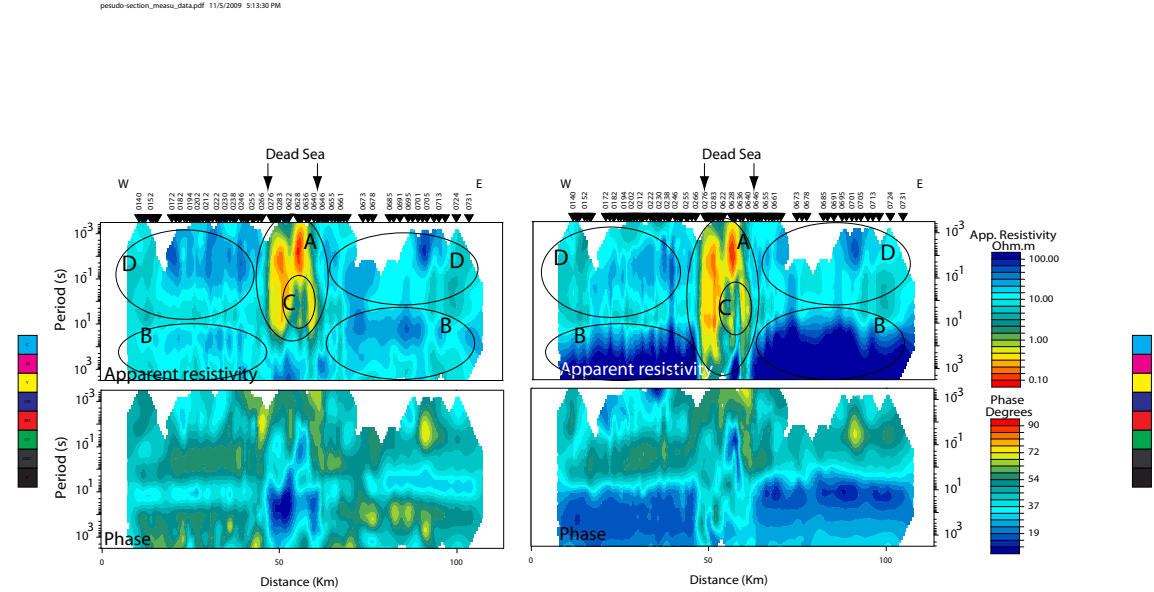


Figure 4.13: Pseudo-sections of apparent resistivity (upper panels) and phases (lower panels) for a) TE-mode and b) TM-mode. Low apparent resistivity values ($< 1 \Omega m$) are observed at stations located in the Dead Sea area. Data from stations located to the east and west of the Dead Sea show a split between TE- and TM-mode at periods around 10s. The TM-mode apparent resistivity values increase and reach values higher than $100 \Omega m$ at periods $> 1000s$, while the corresponding TM-mode phases decrease to $< 10^\circ$.

This first data analysis based on comparison of adjacent stations along the profile using the pseudo-sections (Fig. 4.13) reveals common characteristics for neighbouring sites which allows a classification of the stations into three categories:

- Stations located along the western segment of the profile show relatively high apparent resistivity values ($> 10 \Omega m$) in the short period range ($< 10s$) for both TE- and TM-mode (D). The ρ_{TM} values increase ($> 100 \Omega m$) at period $> 10s$, while the ρ_{TE} values remain at $10 \Omega m$ (B). Such a behaviour is typical for a sequence of resistive-conductive-resistive structures.
- Towards the central part of the profile (Dead Sea area) the apparent resistivity values for both modes start to decrease and reach its minimum of about $0.1 \Omega m$ at shorter periods ($< 1s$) for stations located directly near the Dead Sea and over the Al-Lisan peninsula (A). At periods longer than 10s the ρ_{TE} values

start to increase ($> 10 \Omega m$) while the ρ_{TM} values remain below $1 \Omega m$. This characteristic behaviour for both modes is an indication for a conductor that extends deep into the Earth. Some stations located directly on the Al-Lisan peninsula show relatively high apparent resistivity values ($> 10 \Omega m$) for both modes (C).

- Data from stations located on the eastern part of the profile show, to some extent, the same behaviour as the data from stations on the western part. Except, that the apparent resistivity values in the short and long period ranges for both modes are clearly higher than the apparent resistivity values on the western part. Again, the data indicate resistive-conductive-resistive structures.

4.5.2 Static shift

Static shift can be a major problem when dealing with measured data. This effect is caused by near-surface heterogeneities. From a physical point of view, the diffusion equation derived in section 2.1 (eq. 2.22) contains two terms on the right hand side. The first term describes the inductive effect of the electromagnetic waves and its time-dependency. The second term, however, describes the non-inductive effect due to an accumulation of the electric charges at the conductivity discontinuities as a result of the conservation of the electric field. Since the conservation of the electric field is time-independent, this holds also for the non-inductive 'static' effect. If the dimensions of the near surface heterogeneities are several times smaller than the skin depth, the diffusion equation is dominated by the non-inductive term, because the inductive term decreases with increasing period. This means, a MT station measured close to near surface heterogeneities is dominated by the non-inductive effects. If the heterogeneities are more conductive than the surrounding material, the total electric field is reduced (current channelling). More resistive heterogeneities cause an amplification of the total electric field (current deflection, Jiracek (1990); see Fig. 2.1a and b for details). Since both ρ_{TE} and ρ_{TM} are linearly dependent of the electric field components E_x and E_y , respectively, the ρ_{TE} and ρ_{TM} curves will be downwards shifted in the first case and upwards shifted by a real factor in the second case. The corresponding phases (ϕ_{TE} and ϕ_{TM}) curves, however, are not affected because the static shift effect is time-independent and causes only a reduction or amplification of the total electric field without changing its phase.

Let us now consider some real MT data from stations which contain static shift effects. Figure 4.14 shows apparent resistivity and phases for TE (blue dots) and TM (red dots) of four stations (202, 204, 206 and 210) which are located only about 500 m apart from each other. At all four stations, for periods up to 5s both ϕ_{TE} and ϕ_{TM} curves lie on top of each other, indicating a 1D structures. When examining the ρ_{TE} and ρ_{TM} curves at stations 202, 204 and 206 we can see that up to 5s both curves are

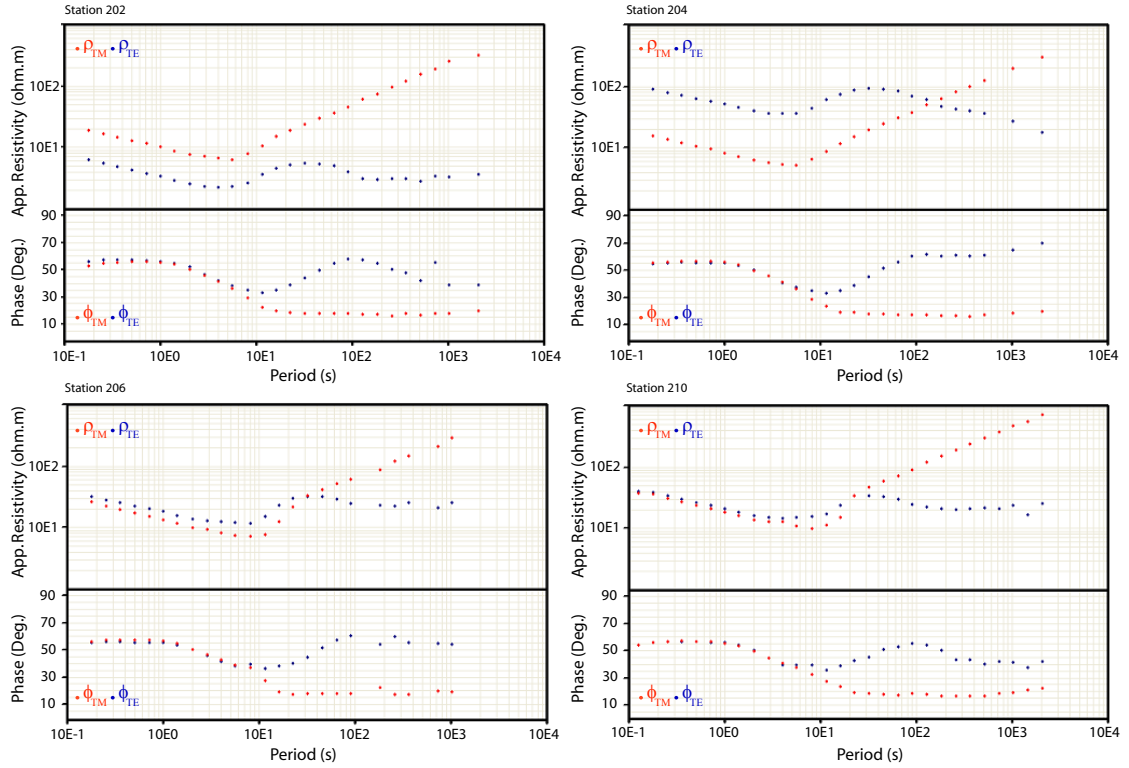


Figure 4.14: Static shift effect at stations 202, 204 and 206 which are located ~ 500 m apart from each other. Station number 210 is used for comparison, since it seems not to be or less affected by static shift. At shorter periods (<10 s) the phases values for TE (blue curve) and TM (red curve) show a typical 1D behaviour. A comparison between the TM-mode apparent resistivity curves at the four stations reveals that the apparent resistivity values at shorter periods (<10 s) do not change too much between the stations. However, the TE-mode apparent resistivity curves are shifted downwards at station 202, upwards at station 204, and slightly upwards at station 206. This implies that the TE-mode apparent resistivity values are more affected by static shift than the TM-mode.

running parallel to each other, but they are shifted from each other. If we compare the values of ρ_{TM} of the four stations, we can deduce that these values do not change too much between stations. This could mean that the ρ_{TE} curves are shifted with respect to the ρ_{TM} curves. At stations 202 the ρ_{TE} curve is shifted downwards by a factor of 3.3 from the ρ_{TM} curve, while at station 204 the ρ_{TE} curve is shifted upwards by a factor of 0.19. At station 206, the static shift factor is around 0.9. Station 210 appears to be less affected by static shift.

4.5.3 2D Inversion

For 2D inversion, I used the regularized non-linear conjugate gradient (NLCG) algorithm in the model space described by [Rodi & Mackie \(2001\)](#) which is summarised in section 2.6. This algorithm is implemented in the WINGLINK software package which offers many options to modify parameters that control the inversion and the forward modelling.

Since the inversion of MT data is a non-linear problem (see section 2.6), a number of electrical conductivity models exists which can explain the measured data to some extent. Thus, considerable effort and care should be taken when inverting the MT data. This requires, for example, exploring the model space in which we are seeking the conductivity distribution. The examination of the model space starts by analysing the influence of the parameters that control the inversion and the forward modelling. Furthermore, known features like oceans or geological structure, particularly if they are laterally displaced from the profile, must be considered and included as a-priori information, if they have an effect on the data.

The accuracy of the computed electromagnetic (EM) solution in the forward modelling depends mainly on the discretization of the subsurface, typically using a 2D staggered grid. Gradient based inversion schemes (Gauss-Newton, Occam and NLCG), require accurate EM solutions to ensure accurate computation of the sensitivity matrix (\mathbf{J} ; See section 2.6). The dimension of the cells in the 2D grid must be kept small compared to the skin depth ([Weaver, 1994](#)) to ensure a good approximation of Maxwell's equations and to reduce numerical instability. However, keeping cell dimensions small causes a huge number of model parameters for which we have to solve both the forward modelling and the inverse problems. Therefore, it is worthwhile to carry out a grid dependency solution analysis to find a trade-off between an accurate solution and optimum run times. This can be achieved by running the forward modelling for simplified 2D models based on the pseudo-sections. The test starts from a coarse grid and successively decreases the cell sizes in the grid. Optimum cell sizes are found if the computed responses do not change even when choosing finer cells dimensions.

Off-profile features

Due to the diffusive nature of the electromagnetic waves in the Earth, features that are laterally displaced from the profile can have significant influence on the measured data. At each MT station we observe the responses of conductivity contrasts which are located in the range of the skin depth (induction volume). Therefore, MT soundings are considered to be a volume rather than pure depth soundings. Laterally displaced structures may be located in profile direction or off profile. In the first case, 2D forward modelling might be still sufficient to analyse the influence of such structures. In

the second case, 3D modelling is required.

In the study area there are two main features that could have an influence on the MT data. The first feature is the Mediterranean Sea which is located ~ 100 km away from the westernmost station (Fig. 4.5). Particularly, the high conductive sea water ($0.3 \Omega m$) can have a measurable effects. The other feature which is very close to the profile are the Dead Sea brines which extent to the north of the study area. This extremely saline water body is up to 300 m deep, 15 km wide and approximately 50 km long (Fig. 4.5). The effect of this feature must also be considered, but to study the influence of this off-profile structure require 3D modelling.

To study the effect of the Mediterranean Sea, I used a simple 2D conductivity structure which consists of a 1D background resistivity with two layers (Fig. 4.15b). The uppermost layer has a resistivity value of $20 \Omega m$ and a thickness of ~ 16 km, whereas the underlying half space is resistive with $1000 \Omega m$. Hundred kilometres away from the westernmost station, a conductive structure ($0.3 \Omega m$) is embedded in the uppermost layer to simulate the Mediterranean Sea. The thickness of this conductive feature is ~ 2.5 km and extends laterally to the left boundary of the model (Fig. 4.15b).

The effect of the Mediterranean Sea on the data (ocean effect) is represented as a splitting of apparent resistivity and phases curves between TE and TM modes (Fig. 4.15a). The apparent resistivity and phases curves of the three selected stations indicate a gradual decrease of this effect along the profile towards the east. At the westernmost station (140) the ocean effect is visible at periods longer than 600s. At the easternmost station (731) a splitting in the apparent resistivity curves between TE and TM appears only at periods longer than 1000s. Whereas, splitting in the phases curves can be observed at periods > 700 s.

Similarly to the behaviour of the apparent resistivity and phase curves, the induction vectors (Fig. 4.15c) reflect the influence of the Mediterranean Sea, however, only at periods > 1000 s. The induction vectors for the two selected periods (1000s and 1585s) in Fig. 4.15c show that at 1000s only stations at the westernmost end of the profile indicate a conductor to the west of the profile.

Regularization parameter

Since the NLCG code of Rodi & Mackie (2001) is a regularized inversion scheme, the regularization parameter τ has a considerable effect on the computed models. As pointed out in section 2.6, a suitable value of the regularization parameter ensures that neither the data misfit term nor the smoothing term dominates the penalty function² (eq. 2.64). Since the search for an optimum value of the regularization parameter is

²Please notice that the parameter τ is the same as λ used in eq. 2.64. However, to be consistent with Rodi & Mackie (2001), τ is used here.

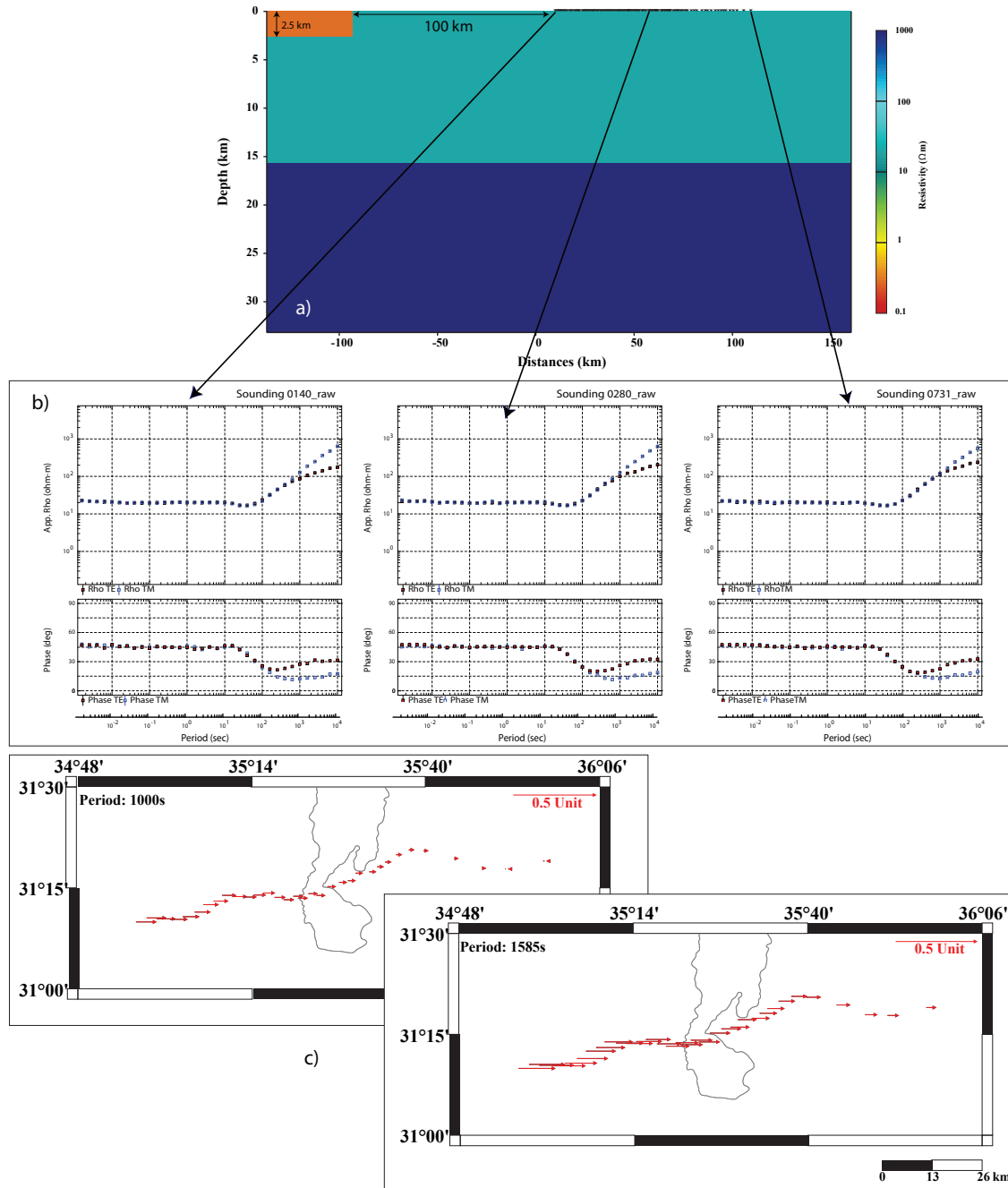


Figure 4.15: a) The synthetic conductivity model used to evaluate the influence of the Mediterranean Sea on the measured profile data. The upper most layer has a resistivity value of 20 Ωm and the lower half space has a resistivity value of 1000 Ωm . 100 km away from the westernmost station, a 2D conductive structure (0.3 Ωm) is embedded in the upper most layer. b) The computed model responses in terms of apparent resistivity and phase at three stations. The influence of the Mediterranean Sea appears in the apparent resistivity and phase curves in form of splitting of the TE and TM apparent resistivity and phase curves at periods longer than 700s and 200s, respectively. c) The real parts of the induction vectors for two periods of 1000s and 1585s. Only at periods longer than 1000s, the influence of the Mediterranean Sea is visible in the induction vectors. As expected, the influence is most obvious at the western stations.

not part of the NLCG code it must be found ‘manually’. This is achieved by running the inversion using several values of τ and choosing the one that ensures both good data fit and a smooth model.

To determine a suitable value of τ , I used the ρ_{TM} , ϕ_{TM} , ϕ_{TE} and T_{zy} for the 2D inversion while ρ_{TE} was down-weighted using a high error floor to avoid problems with the static shift. Six regularization values ($\tau=0.1, 1, 20, 50, 100$ and 1000) were tested using a homogeneous half space of $100 \Omega m$ as a starting model for all inversion runs. Figures 4.16a, b and c show three models representing low ($\tau=0.1$), intermediate ($\tau=20$) and high ($\tau=1000$) smoothing, respectively. Figure 4.16c illustrates the resulting L-curve as a comparison between data misfit and model roughness.

As discussed in section 2.6, a low value of the regularization parameter τ causes

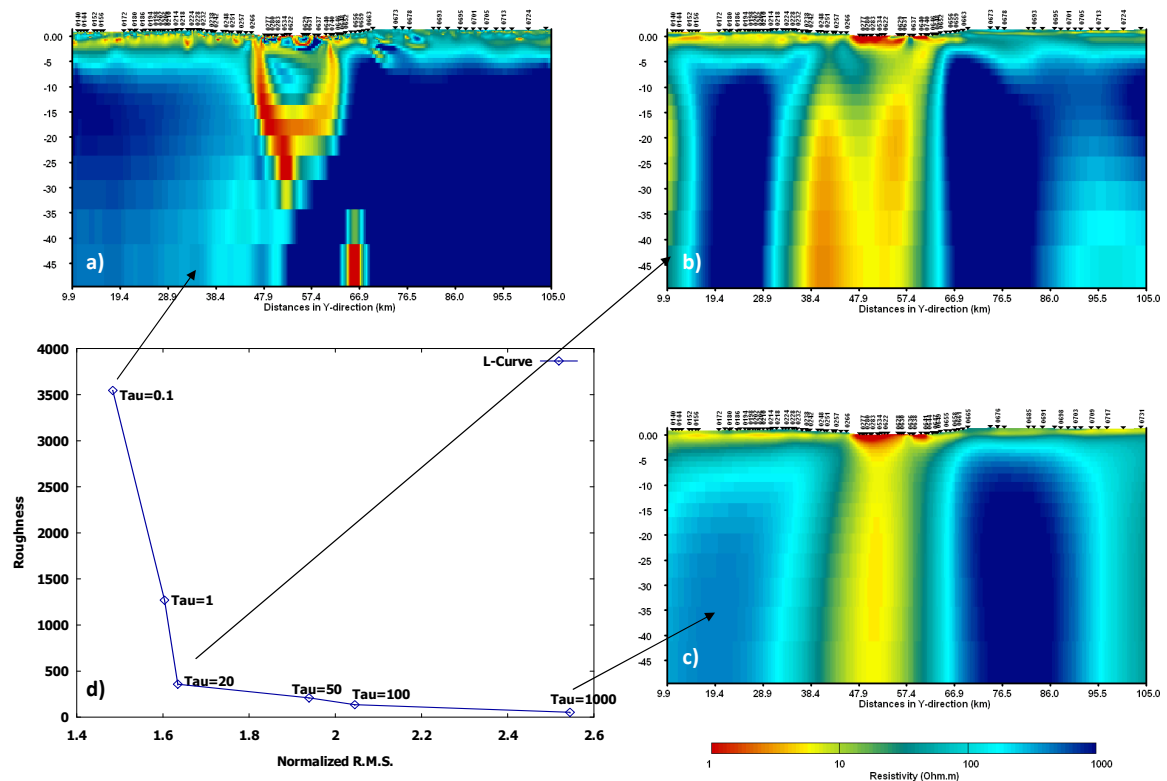


Figure 4.16: Inversion tests to determine the optimum value of the regularization value τ . a) b) and c) are three models computed using values 0.1, 20 and 100 of τ , respectively. The extreme value of $\tau = 0.1$ produces a very rough model with good data fit (small RMS value). On the other hand, the use of $\tau=1000$ in the inversion produces a very smooth model but worse data fit (high RMS value). d) The relationship between the normalised RMS and the model roughness takes the form of an L-curve. The optimum value of τ corresponds to the maximum curvature of the L-curve, which in this case is close to $\tau = 20$.

that the penalty function is dominated by the data misfit term while the roughness

term is almost neglected. The high roughness of the model in Fig. 4.16a shows disconnected structures especially in the upper part of the model (up to 5km in depth). High value of the regularization parameter ($\tau=1000$) causes that the roughness term in the penalty function has more weight and the data misfit term is down-weighted (Fig. 4.16b). In this case the resulting model is very smooth and the structures are more connected due to the required smoothness of model. However, the data misfit becomes significantly worse (Fig. 4.16c), because small structures that are required by the data have been smeared out and integrated with bigger structures to ensure high smoothing. An intermediate value of the regularization parameter ($\tau=20$) ensures that the data misfit and the roughness terms are well balanced in the penalty function (Fig. 4.16b). The model which corresponds to $\tau=20$ has a normalised RMS value equal to 1.63 and roughness equal to 358.4. These two values correspond to the maximum curvature of the L-curve and the associated regularization parameter is considered to be the optimum value for the data set under consideration (Fig. 4.16d). A comparison between the three models in Fig. 4.16 shows that there are three common structures that seem to be independent from the selection of the regularization parameters. Two resistive blocks that start at a depth of ~ 5 km are separated in the central part of the model. The central part of the model always contains very conductive structures ($<1 \Omega m$) which extend ~ 10 km in horizontal direction and to at least ~ 30 km in depth.

Influence of the individual transfer functions

In the previous inversion test to determine an optimum value of τ , all available data have been used. Consequently, it is difficult to assess which structures are required by which data components. Therefore, it is more instructive to run the inversion using individual data components.

From a physical point of view, it seems advantageous to start exploring the model space by inverting a transfer function which is less affected by off-profile structures and less liable to static shift effect (Favetto *et al.*, 2008). In a 2D Earth, the vertical magnetic field component B_z is caused by electric currents induced in the Earth which flow parallel to the strike direction (E-polarisation). Therefore, the only vertical magnetic transfer function that can be fitted in a 2D inversion is T_{zy} . Since T_{zy} is particularly sensitive to lateral changes in the conductivities, it can be inverted separately to detect vertical discontinuities in the model.

To invert the T_{zy} data, a homogeneous half space of $100 \Omega m$ was used as a starting model. The error floor was set to 0.03 of the absolute values of T_{zy} and a regularization parameter $\tau=20$ was used. Figure 4.17 shows the inversion result fitting only the

T_{zy} component after 500 iterations with a global RMS value³ of 1.6.

The model in Fig. 4.17 shows that the main features are recovered. The inversion

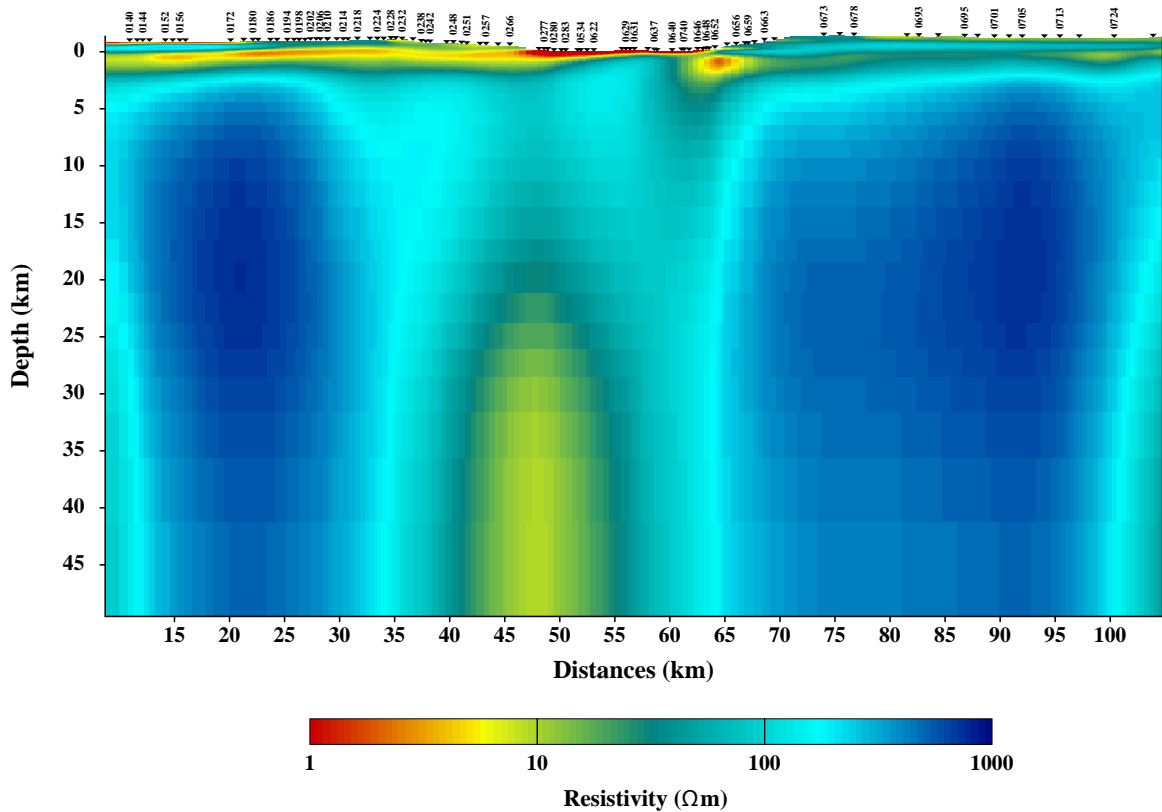


Figure 4.17: The inversion result of fitting only the T_{zy} data. In the central part of the model at a depth of ~ 25 km, a vertical conductor ($\sim 10 \Omega m$) appears in the model. Two high resistive blocks are located to the east and west of this conductor. Other vertical discontinuities can be observed in the shallow part of the model between station 266 and 277 and between station 652 and 659.

result fitting only T_{zy} confirms the existence of a vertical conductor in the central part of the profile starting at a depth of ~ 20 km. This feature represents a sharp horizontal transition between the western and eastern resistive blocks and the central conductive zone. However, the top of the of the central vertical conductor and the western and eastern resistive blocks are not well resolved, due to the low resolution of T_{zy} to detect layered features. The model in Fig. 4.17 shows another vertical discontinuity at shallow depth between station 266 and 656.

To assess the data fit for the model in Fig. 4.17, a comparison between the measured and predicted data in terms of pseudo-section of the real and imaginary part of the T_{zy} data is presented in Fig. 4.18.

³The global RMS value refers to the RMS value calculated for all periods, all stations and all data components. Total RMS definition will be used later to refer to the RMS value calculated for all periods at each station and for all data component.

For the western and eastern stations, there is good agreement between measured and

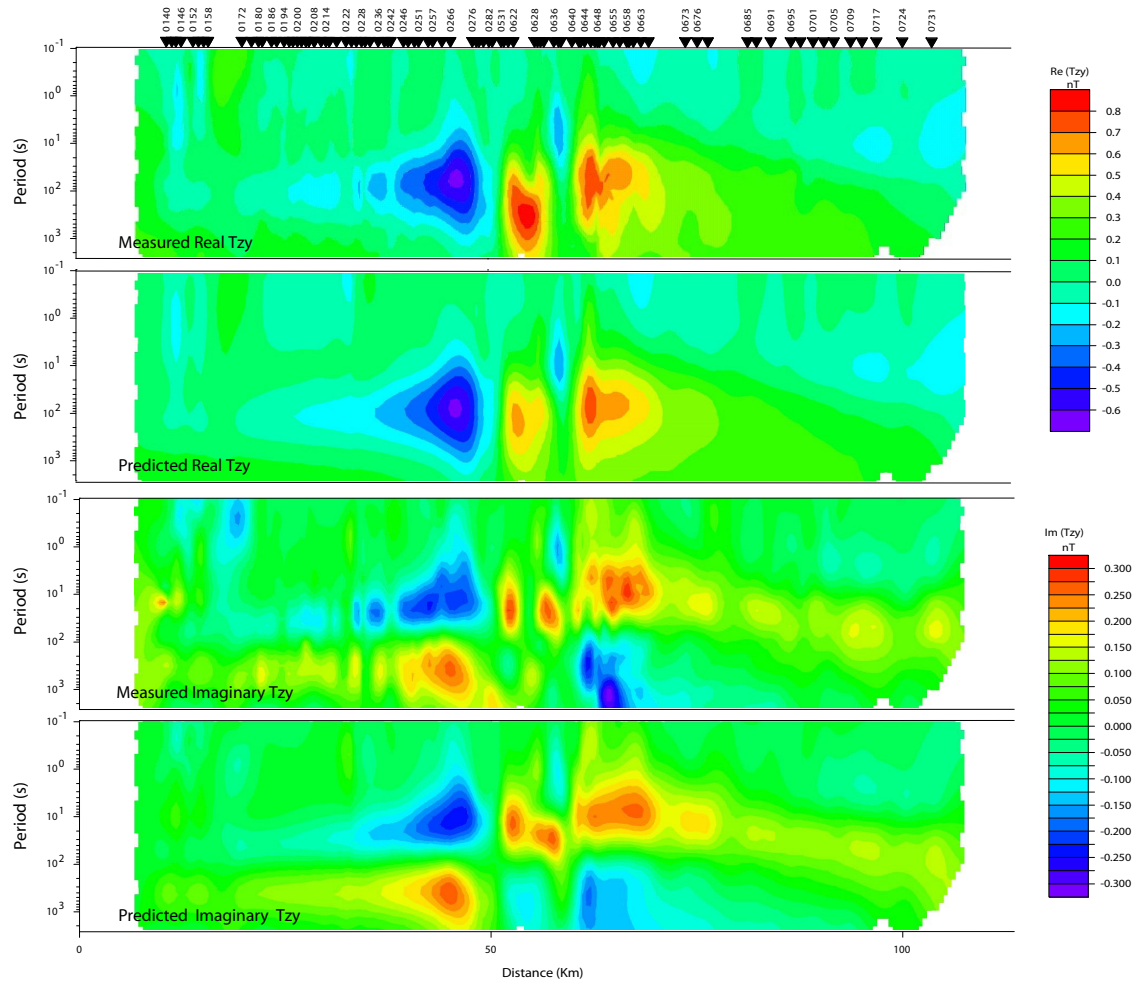


Figure 4.18: A comparison between the real (two upper panels) and imaginary (two lower panels) part of the measured and calculated T_{zy} data. This visual comparison reveals a good data fit between measured and calculated data within the given error bounds.

predicted real and imaginary parts of T_{zy} data. Only a few stations at the western part show a discrepancy between the measured and predicted imaginary part of T_{zy} , in particularly at short periods ($< 1s$). Stations located in the central part of the profile show relatively good agreement between the measured and predicted data for both, real and imaginary part of T_{zy} . However, the amplitude of the real part of T_{zy} is not reached by the inversion. This discrepancy is most obvious for positive values of the amplitude (red colour in Fig. 4.18) at periods between 100s and 1000s. This can also be observed for the imaginary part of T_{zy} . The slightly poorer data fit for both, real and imaginary part of T_{zy} at the central stations is probably due to the complexity of structures in this region.

The second transfer function to be inverted separately is the TM-mode. The 2D inversion of the TM-mode data was obtained with the same grid spacing as used for inverting the T_{zy} component and the same regularization parameter ($\tau = 20$). The starting model was a homogeneous half space with electrical resistivity of $100 \Omega m$. Error floors of 5% and 2% were set to ρ_{TM} and ϕ_{TM} , respectively. The 2% error of the phases is equivalent to 0.6° . Figure 4.19 shows the inversion result after 2000 iterations with a global RMS of 1.55.

The model in Fig. 4.19 shows that the high resistive blocks on both sides of the pro-

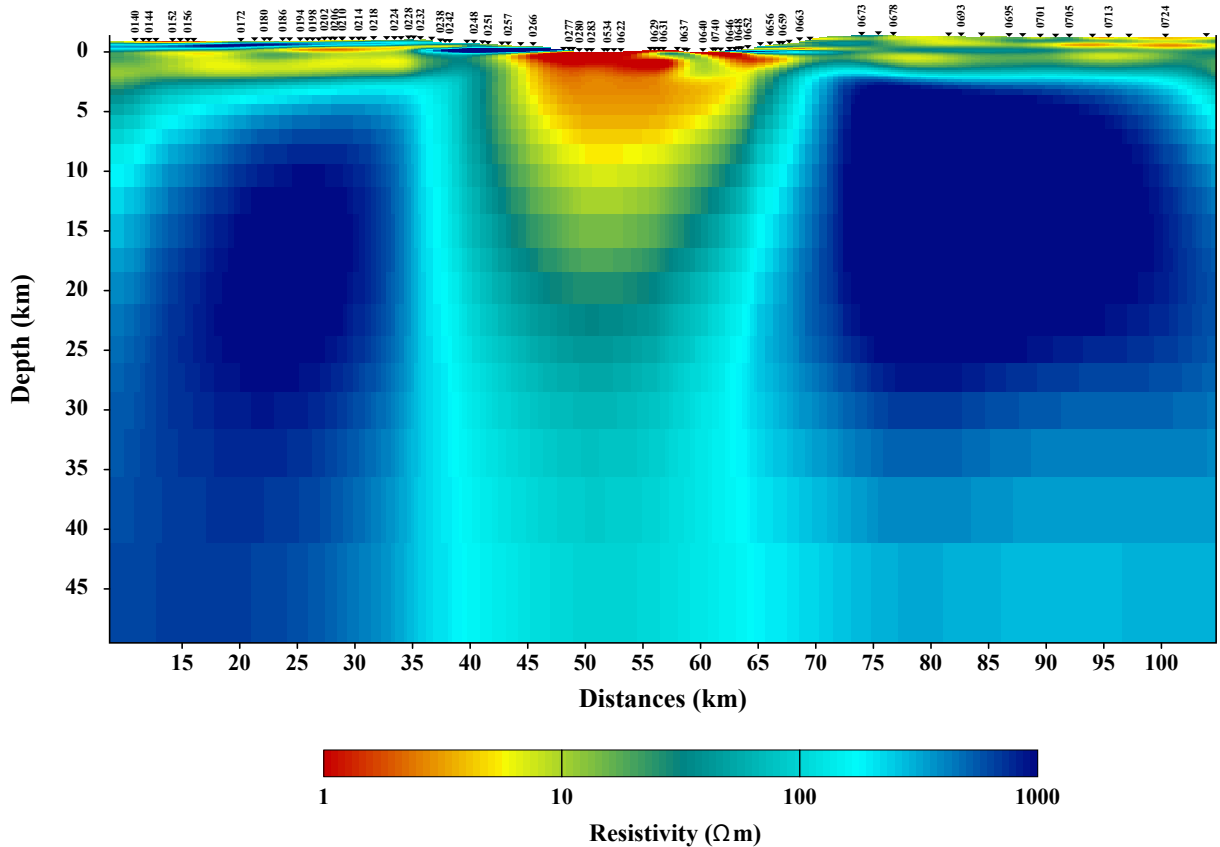


Figure 4.19: The inversion result of fitting only the TM-mode data. The central part of the model reveals a good conductor extending to a depth of ~ 15 km delimited by resistive blocks towards E and W. The vertical conductor that appears in the central part at a depth of ~ 20 km in the model presented in Fig. 4.17 is absent. However, several layered structures appear in the upper 4 km in the model.

file (western and eastern blocks) appear again when fitting only the TM-mode data. The top of the two blocks is now well resolved due to the high resolution of TM-mode data in detecting conductivity contrast in vertical direction. In contrast, the vertical conductor that appears in the central part at a depth of ~ 20 km in the model presented in Fig. 4.17 is absent in Fig. 4.19. Instead, a conductive structure appears in the central part of the model which is delimited by the western and eastern resistive

blocks extending to a depth of ~ 15 km. The resistivity values inside this conductor increase gradually from $< 1 \Omega m$ near the surface to $\sim 10 \Omega m$ at a depth of ~ 15 km. In the upper part of the model (up to ~ 4 km), sequences of conductive-resistive layers at the western and eastern segments are more pronounced than in Fig. 4.17.

Another data set which is available for the 2D inversion is the TE-mode data in which the electric currents flow parallel to the strike direction. Due to the fact that the electric currents flow parallel to the strike direction, off-profile and 3D structures have significant effect in these data type (Weaver, 1994; Simpson & Bahr, 2005). Furthermore, in the DESIRE-MT data set, the TE-mode data appear to be more affected by static shift than the TM-mode data (as discussed in Fig. 4.14). For these reasons, considerable care must be taken into account when inverting the TE-mode data in 2D. Particularly, off-profile feature may force the 2D inversion to construct structures at depth which are originally displaced from the profile along strike.

Two inversion runs were carried out using the TE-mode data. In the first inversion run, similar grid spacing, regularization parameter and error floor values were used as for the TM-mode inversion. The inversion stopped after 350 iterations with an RMS value of 2.8. Figure 4.20a shows this inversion result using error floor value of 5% for ρ_{TE} and 2% for ϕ_{TE} . A comparison with the inversion result obtained using TM-mode data (Fig. 4.19) shows that up to a depth of ~ 3 km, similar structures appear in both models. However, in addition, four conductive structures in form of blobs at depths of ~ 5 km are required by the model to fit the TE-mode data within the error bounds. It is worthwhile noting that the two inner conductive blobs are located symmetrically around the central part of the model which appears to be conductive when inverting for T_{zy} and TM-mode data separately (Figs. 4.17 and 4.19).

In the second inversion run of the TE-mode data, a high error floor (1000%) was assigned to ρ_{TE} , so that effectively only ϕ_{TE} is used in the inversion. Figure 4.20b shows that when inverting only the ϕ_{TE} data, similar structures appear as those obtained when fitting T_{zy} and TM-mode data separately. Particularly, the western and the eastern resistive blocks and the vertical conductor in the central part appear in the model. For this inversion test, 1000 iterations were required to reach a much better RMS value of 1.18.

A comparison between the measured and calculated data for the model in Fig. 4.20a, shows that data fit for both ρ_{TE} and ϕ_{TE} is acceptable (Fig. 4.20c). However, the structures produced, in particular the conductive blobs can not be interpreted and they contradict the computed models to fitting the T_{zy} and the TM-mode data. Another important criterion is to consider the fit of the ϕ_{TE} data in both models. By visual comparison between the calculated ϕ_{TE} data in Fig. 4.20c and d (lower panels) with the measured ϕ_{TE} data, we can deduce that the calculated ϕ_{TE} in Fig. 4.20d are better fitted, particularly for those stations located in the central part of the profile. To be more precise in taking this criterion in consideration, a comparison between

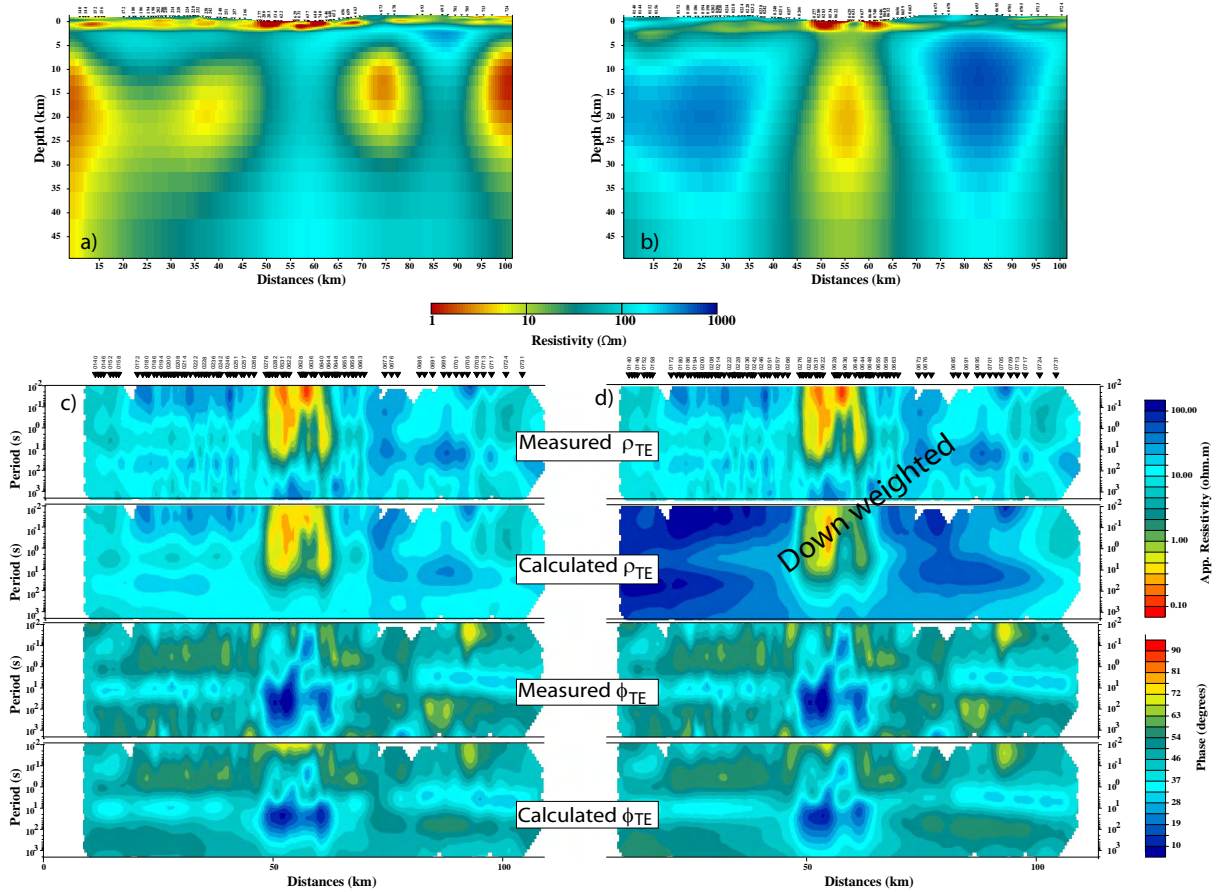


Figure 4.20: Two inversion results of fitting a) ρ_{TE} and ϕ_{TE} , b) ϕ_{TE} and down-weighted ρ_{TE} by using high error floor (1000%). c) and d) are pseudo sections for the measured and computed ρ_{TE} and ϕ_{TE} corresponding to the 2D models in a) and b), respectively. The inversion result of fitting the ρ_{TE} and ϕ_{TE} data (a) shows that additional structures (four conductive blobs) appear in the model which are incompatible with the previous inversion results of fitting T_{zy} and TM-mode data separately. However, the model obtained by fitting only the ϕ_{TE} data (b) shows, to some extent, similar structures as obtained previously. Moreover, the calculated ϕ_{TE} pseudo section in d) fits the measured data better than the calculated ϕ_{TE} pseudo section in c).

the RMS values for both models is revisited. The global RMS value of 1.18 for the model in Fig. 4.20b corresponds only to ϕ_{TE} , since ρ_{TE} is down weighted. Taking the model in Fig. 4.20a, similarly down weighting the ρ_{TE} data and then computing the model responses shows that the obtained global RMS value of 1.42 is significantly worse than 1.18.

The inversion results presented in Figs. 4.17, 4.19 and 4.20 to fit T_{zy} , TM- and TE-mode data individually, show that each data set has different ability to manifest either vertical or horizontal electrical discontinuities. If the data are 2D, all components must be fitted with one model. Which means that a 2D joint inversion of different data set should give a robust image of the conductivity distribution of the subsurface.

Since, vertical and horizontal discontinuities are best resolved using the T_{zy} and the TM-mode data, respectively, these two data sets are used first for the joint inversion. The model in Fig. 4.17 is used as starting model. The same error floors and regularization parameter are used for the joint 2D inversion of T_{zy} and TM-mode data. Figure 4.21 shows the resulting conductivity model.

The model in Fig. 4.21 shows that the vertical as well as the horizontal boundaries of the western and eastern resistive blocks are well resolved. Furthermore, the top of the vertical conductor that appears at a depth of ~ 20 km seems to be connected with shallower conductive structures. The most conductive part of this vertical conductor is now located at depths between 25 km and 40 km. The highly conductive structure in the central upper part which is located between station 257 and 663 and extending to a depth of ~ 20 km in Fig. 4.19, is restricted laterally between stations 277 and 652 in Fig. 4.21. Moreover, a vertical differentiation in resistivity values is observed between very high conductivities ($< 1 \Omega m$) extending to a depth of ~ 3 km and reduced conductivities ($\sim 5 \Omega m$) underlying structures extending to a depth of ~ 10 km. At the western and eastern segments down to a depth of ~ 5 km, similar layered structures can be observed as in Fig. 4.19. However, more lateral discontinuities can also be observed.

Figure 4.22a shows a comparison between the normalised RMS values calculated at each station along the profile when fitting the T_{zy} data separately (red curve) and jointly with the TM-mode data (blue curve). It is obvious that inverting T_{zy} and TM-mode data jointly causes an increase of the total RMS values, particularly at those stations located between 50 km and 100 km. This is because; additional layers that are required to fit the TM-mode data have been included in the upper 5 km of the model. The comparison of the RMS values between inverting TM-mode data separately (red curve in Fig. 4.22b) and jointly with T_{zy} data (blue curve), shows slightly increased total RMS values along the profile.

In general, the RMS values of the T_{zy} data are more scattered along the profile in comparison to the RMS values of the TM-mode data. Moreover, high RMS values

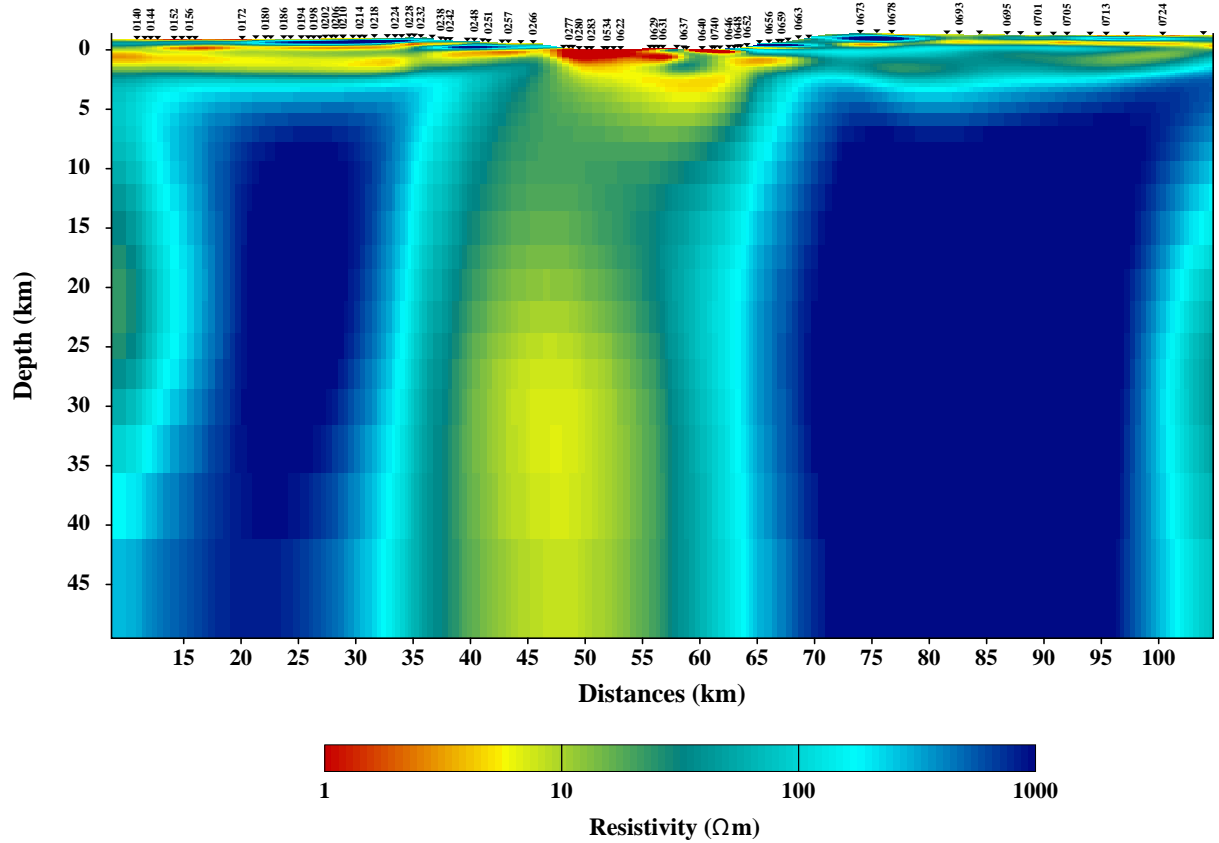


Figure 4.21: The inversion result when fitting the T_{zy} and the TM-mode data jointly. The model shows that the horizontal as well as the vertical conductivity contrasts are well pronounced. For example, the deep vertical conductor in the central part of the model which is required by the T_{zy} data appears together with the central upper part conductor required by the TM-mode data.

can be observed at stations located in the central part of the profile which could mean that the 2D inversion cannot find a suitable model in this region because of 3D effects present in the data.

The inversion result of the TE-mode data showed that only the ϕ_{TE} data can be used for the 2D inversion (Fig. 4.20). Furthermore, the model presented in Fig. 4.20b shows that the vertical conductor in the central part appears enhanced when fitting only ϕ_{TE} . Therefore, for the next inversion run, T_{zy} , ρ_{TM} , ϕ_{TM} , and ϕ_{TE} data were used to find a model that can fit all available data. The same grid spacing and regularization parameter ($\tau = 20$) are used. The error floors were set to 0.03 for T_{zy} , 5% for ρ_{TM} and 2% for both ϕ_{TM} , and ϕ_{TE} . The model presented in Fig. 4.21 was used as starting model. 2000 iterations were required to reach a global RMS value of 1.81 for the model in Fig. 4.23

The model in Fig. 4.23 recovers all structures that have been found when inverting the various individual components of the MT data. Moreover, including the ϕ_{TE} data

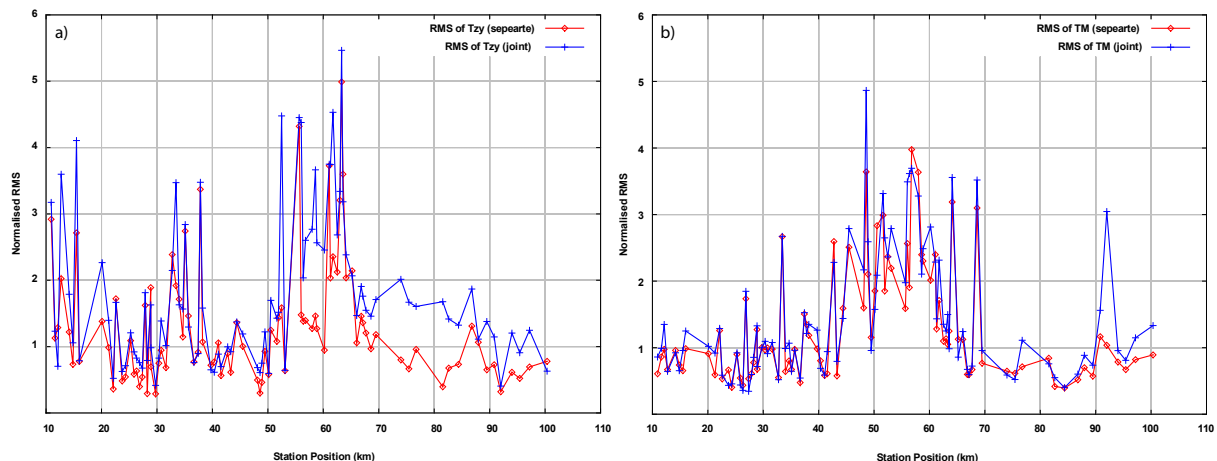


Figure 4.22: a) Comparison between the fit of the T_{zy} data inverted separately (red curve) and jointly with the TM-mode data (blue curve). b) Comparison between the fit of the TM-mode data inverted separately (red curve) and jointly with the T_{zy} data (blue curve). It is obvious in (a) that inverting the T_{zy} and TM-mode data jointly results in a poorer fit for the T_{zy} data. However, the joint inversion of the T_{zy} and TM-mode data does not effect too much the fit of the TM-mode data (b).

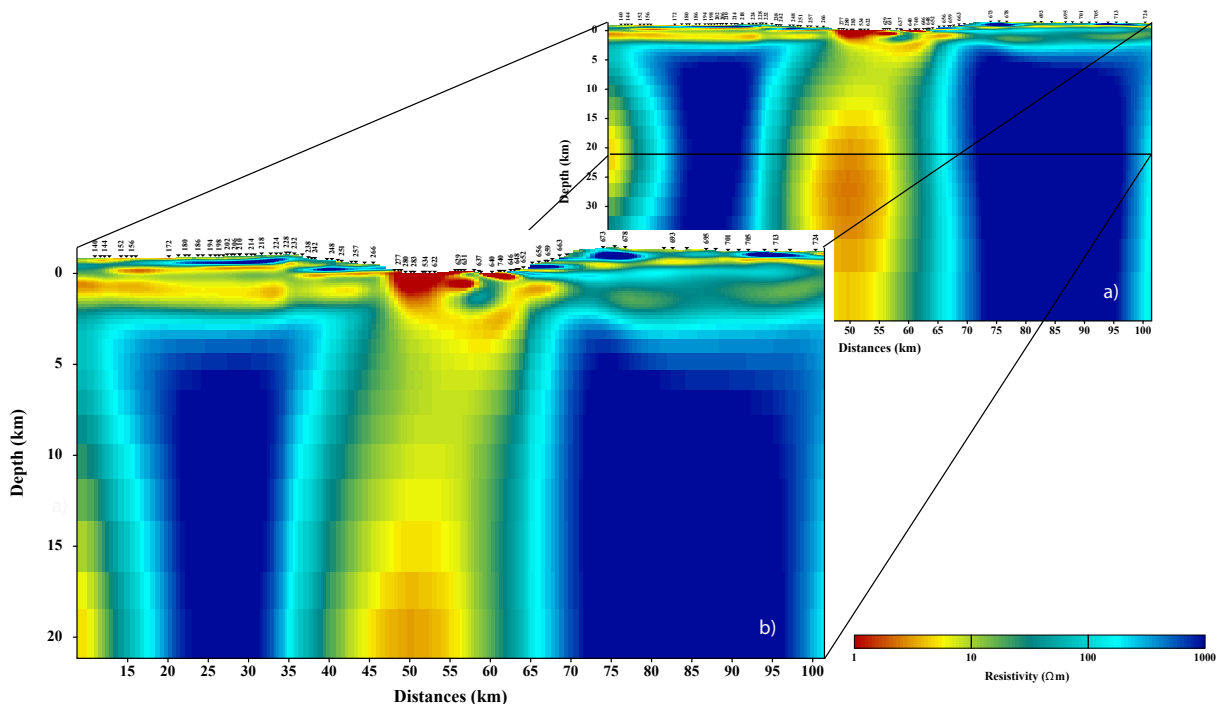


Figure 4.23: The obtained 2D model using all available data (T_{zy} , ρ_{TM} , ϕ_{TM} , and ϕ_{TE}) for the 2D inversion. Fitting the ϕ_{TE} data jointly with the T_{zy} and TM-mode data pronounce the conductive regions in comparison to the model obtained in Fig. 4.21.

for the 2D inversion pronounced the conductive regions in the model, i.e. the central vertical deep conductor and the central shallow conductor.

To evaluate the data fit of the model in Fig. 4.23, the RMS values were calculated for each data type separately at each station along the profile (Fig. 4.24).

Figure 4.24a shows the distribution of the total RMS values along the profile. For the

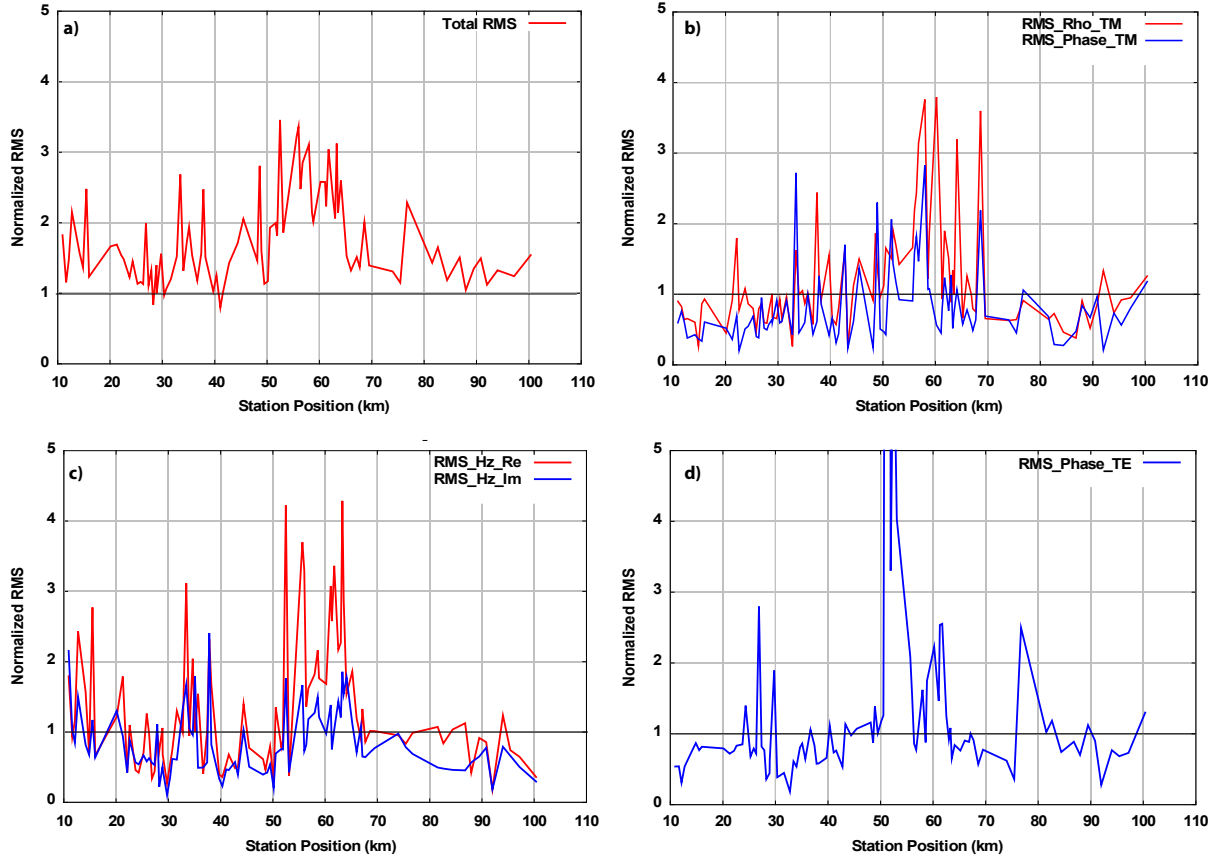


Figure 4.24: The computed RMS values for the model in Fig.4.24. a) The total RMS values at each station containing all data components used for the 2D inversion. The RMS values for the individual data components b, c and d show that an acceptable data fit is reached within the used error floors. Outliers in the data fit can be observed at some stations at the eastern and western segments of the profile. All RMS curves show a poor data fit for the central stations.

eastern and western stations, an average total RMS value of ~ 1.5 can be observed. At stations located in the central part, the total RMS values vary typically between 2 and 3. Considering the RMS values for ρ_{TM} and ϕ_{TM} in Fig. 4.24b, one can observe that the RMS curve of ϕ_{TM} has lower RMS levels than the RMS curve of ρ_{TM} . This means that the ϕ_{TM} data were better fitted than the ρ_{TM} data. Similar behaviour at the western and the eastern stations as in Fig. 4.24a can also be observed in Fig. 4.24b. However, the RMS values of the western and the eastern stations are typically

below 1, while they vary between 1 and 3.5 for the central stations for ρ_{TM} and ϕ_{TM} . Comparing the RMS values between the real and imaginary parts of T_{zy} (Fig. 4.24c) shows that the imaginary part is better fitted than the real part. This is most obvious at the central stations where the RMS values vary between 1 and 2 for the imaginary part and between 2 and 4 for the real parts. The RMS level at the eastern and the western stations lies around 1. The RMS values for ϕ_{TE} (Fig. 4.24d) at the western and the eastern stations are below 1, except at few stations where the RMS have high values.

In summary, the model in Fig. 4.23 fits all available data for the 2D inversion in the given error bound. However, we must keep in mind that some data components are better explained by the model, as the 2D inversion tries to find a model which fits all available data jointly.

Before starting to evaluate the significance of the various conductive structures in the model in Fig. 4.23 by running resolution tests, let us first discuss the potential influence of model boundaries. The 2D models presented so far, did only show the inner parts of the models. The full model extends from -10000 km to 10000 km in horizontal directions and to a depth of 3500 km. Expanding the models to these dimensions is essential for accurately solving the forward model problem as we are dealing with a boundary value problem. The western, eastern and bottom boundaries should be placed far away from any 2D discontinuity, so that the electromagnetic fields at these boundaries can be considered as uniform (Weaver, 1994). On the other hand, expanding the model to such huge dimensions allows the inversion process to place artificial structures in these parts of the model.

Figure 4.25 shows the entire horizontal model extension from ~ -1000 km to ~ 1000 km and to a depth of ~ 500 km. The most obvious structure is the highly conductive ($< 1 \Omega m$) layer at the entire western side of the model and in fact to the western boundary of the model at -10000 km. This layer starts at a depth of ~ 10 km and extends down to ~ 30 km. To the east of the easternmost station exists a huge conductive layer ($\sim 20 \Omega m$) which extends laterally to the eastern boundary of the model at 10000 km. This layer starts at a depth of ~ 16 km and extends vertically down to ~ 200 km. The vertical conductor in the central part of the model can be observed down to ~ 350 km depth.

The obvious question now is, if these structures are required by the data, or if they are inversion artifacts. Answering these questions requires resolution tests to determine the depth and horizontal extents for which the data are still sensitive to changes in the model parameters.

One possibility to verify the depth resolution of our data set is by successively terminating the bottom of the inverted model with a homogeneous half space (e.g. $1000 \Omega m$), computing the model responses and observe the changes in the data misfit. As a criterion for significant changes in the model responses, we may use the global

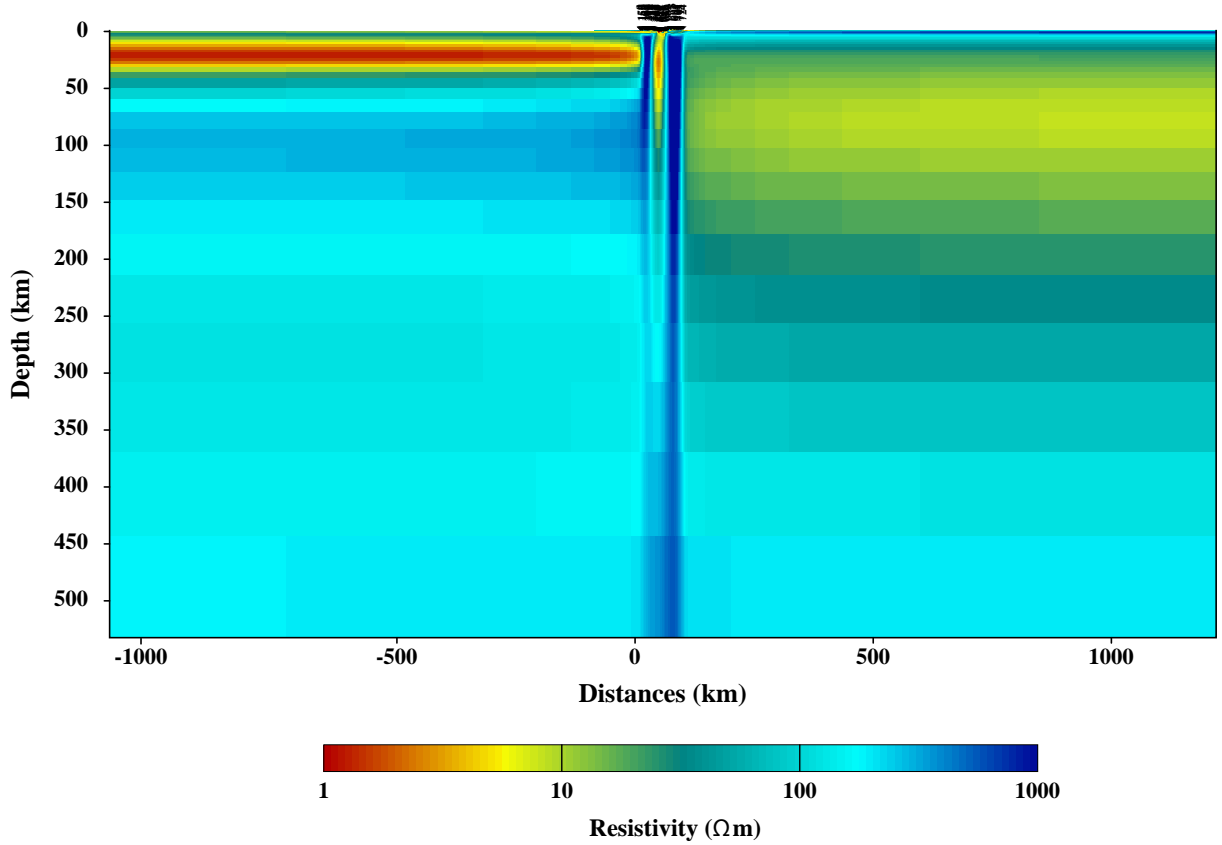


Figure 4.25: An extended view of the model in Fig. 4.23 showing the structures from ~ -1000 km to ~ 1000 km in Y-direction and to a depth of ~ 500 km. The most obvious structure is the high conductive ($< 1 \Omega m$) layer at the entire western side of the model which extends further to the western boundary of the model.

RMS value, which represents the normalised data misfit. The global RMS value is computed for all stations and all periods. However, we are more interested to know which stations are influenced by a certain feature in the model. A better approach is to use the differences between the pseudo-section of the model that was not terminated (Fig. 4.25) and the pseudo sections for models that are terminated with a homogeneous structures. This kind of computation is similar to the sensitivity matrix computation discussed in section 2.6.2 eq. 2.73, with the difference that several model parameters are considered and the results will not be normalised with the model differences (Δm).

Starting from a depth of 2000 km upwards to 40 km, 10 models were created by terminating successively the model in Fig. 4.25 with a homogeneous half space ($1000 \Omega m$) at different depths. Two models were used to explain the procedure of determining the resolution depth of the data (Fig. 4.26a and b).

The first noticeable differences in the model responses are observed when introducing a horizontal discontinuity at a depth of ~ 210 km. Due to its high sensitivity in

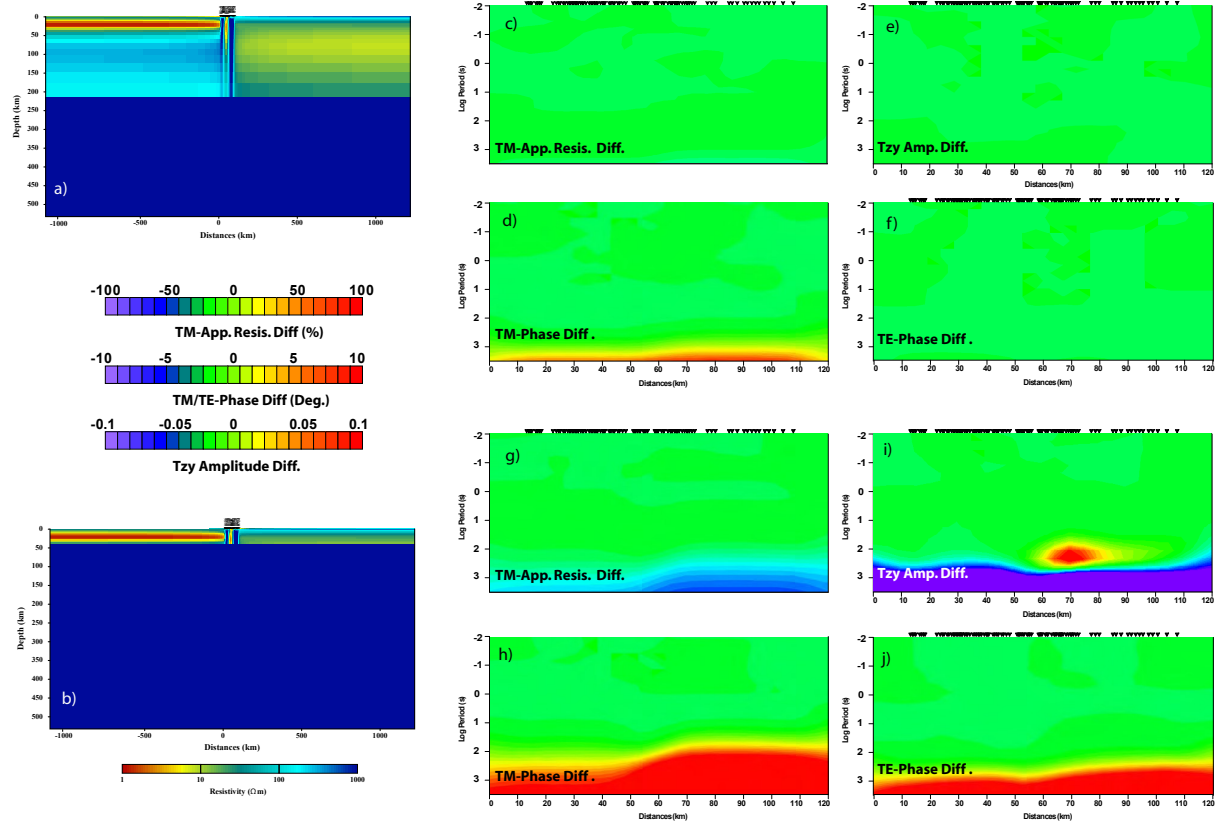


Figure 4.26: Forward modelling study using the constrained 2D models a and b to determine the depth resolution of the measured data. The pseudo section differences (c, d, e, and f) between the model responses in Fig. 4.25 and the model responses of the model in a) indicate that terminating the model at a depth of ~ 210 km has a slight impact on the model responses. On the other hand, terminating the model at a depth of ~ 45 km effects all model responses presented in j, i, h, and g from a period of ~ 100 s and upwards.

detecting horizontal discontinuities, the TM-mode data, in particular ϕ_{TM} , is most influenced by this termination (Fig. 4.26d). At periods longer than 1000s, 5 degrees phase difference between the ϕ_{TM} values calculated for the models in Fig. 4.25 and in Fig. 4.26a can be observed. No significant differences are observed in the other model responses in Fig. 4.26c, e and f. Terminating the model at a depth of ~ 45 km (Fig. 4.26b) has an influence on all model responses as can be seen in Fig. 4.26g, h, i and j. Differences of more than 5 degrees in ϕ_{TM} at periods > 30 s and more than 10% differences in ρ_{TM} can be observed (Fig. 4.26h, g, respectively). The pseudo-section differences of the T_{zy} amplitude (Fig. 4.26i) shows for periods > 100 s that all sites are influenced by terminating the model at a depth of ~ 45 km. The differences in the TE-mode phases (Fig. 4.26j) show a similar pattern, however, at periods longer than 100s. It is also interesting to note that the TM-mode data at the eastern stations are

more influenced than the western stations if the model is terminated at a depth of ~ 45 km.

The depth resolution tests discussed above show that terminating the model at a depth of ~ 45 km has a significant influence on the computed model responses. However, this result does not indicate if this termination improves the data fit, since it has been shown only that there are differences between the computed model responses. A comparison between the computed RMS values for the model in Fig. 4.25 and i.e. for the model in Fig. 4.26b, shows clearly that terminating the model at a depth of ~ 45 km, results in worse data fit which is expressed in high RMS values (Fig. 4.27). As the result in Fig. 4.26h shows and is confirmed in Fig. 4.27b, the TM-mode phases for the constrained model have higher total RMS values than for the unconstrained model.

Similar resolution tests have been carried out to determine the lateral resolution of

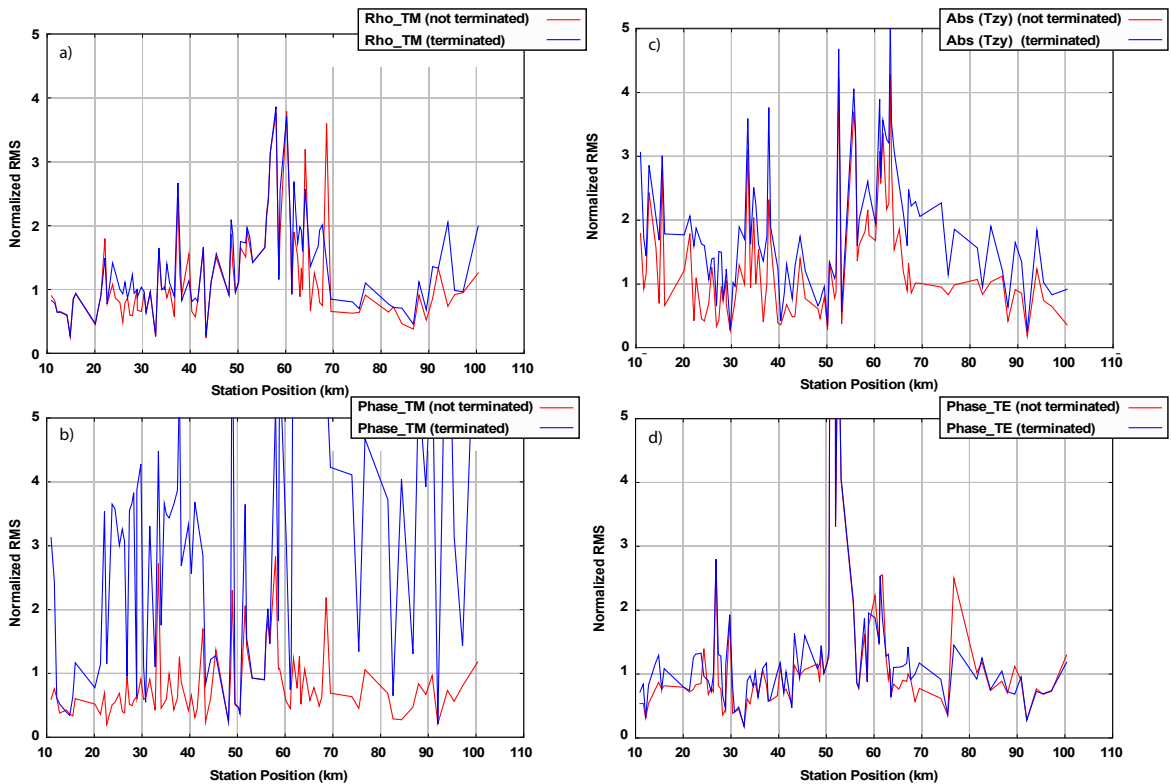


Figure 4.27: Comparison between the RMS values compute for the model in Fig. 4.25 (blue curves) and the constrained model in Fig. 4.26b (red curves). In general, terminating the model at a depth of ~ 45 km worsen the data fit. Particularly effected are the ϕ_{TM} data fit (c).

the data. These tests reveal that the model in Fig. 4.25 can be terminated 200 km west of the westernmost station and 200 km to the east of the easternmost station. Taking these limits into consideration, a new model was created in which the model

parameters beyond these limits were assigned to constant resistivity values and kept fixed during the inversion runs. The upper 3 km at the eastern and western sides could be altered by the inversion. This is to testify if the deep conductor to the west of the westernmost station which appears at a depth of ~ 15 km in the model presented in Fig. 4.25 can be replaced by a shallow conductive layer at the western side. The latter seems to be more realistic for two reasons: First, this conductive layer may be considered as an extension of the western conductive layer that appears at a depth of ~ 2 km in all models presented so far. Second, this shallow conductive layer could coincide with the Mediterranean sea which is located ~ 100 km to the west of the westernmost station.

In the first inversion test with the new constrained model, a resistivity value of $100 \Omega m$ was assigned to all model parameters, including the fixed areas. The same error floors and regularization parameter as before were used for this test. Figure 4.28 presents the inversion result after 1000 iterations with an RMS error of 1.83.

As can be seen, the conductive layer which was previously placed at a depth of ~ 15 km to the west of the westernmost station is replaced by a relatively shallow conductive layer that can be seen as an extension of the deepest conductive layer beneath the western segment of the profile (Fig. 4.28). Comparison of the model in Fig. 4.23b with the model in Fig. 4.28 shows that fixing the boundaries of the model does not influence the structures of the top 20 km. However, the sub-vertical deep conductor in the central part of the model seems to be placed more towards the west. This is because, the western deep conductive layer is now replaced with the western shallow layer. Therefore, the inversion process fits the T_{zy} component of the western stations by extending the central deep conductor further to the west.

To quantify the influence of fixing the boundary region on the data fit, a comparison between the RMS values computed for the models in Figs. 4.23 and 4.28 is shown in Fig. 4.29. In general, the result in Fig. 4.29 shows that fixing the boundary regions of the model does not effect the data fit. However, one can observe small differences in the RMS values particularly at the westernmost and easternmost stations. Figure 4.29 shows that the data fit is improved at some stations, but worsened at other stations. Therefore, it is difficult to asses which model is a better representative of the data. However, the model presented in Fig. 4.28 seems to be more realistic in geological context

After handling the boundary regions of the model where we have no data, we turn our attention to the central part of the model which extends laterally along the profile and to a depth of ~ 30 km. In all 2D models presented so far, we observe: i) an area with strong heterogeneities in the upper crust (< 5 km) and ii) the sub-vertical deep conductor. The obvious questions are: How reliable is this vertical deep conductor? Is it really required by the data, or just an inversion artifact. If this conductor is required by the data, to which depth does it extend? To Answer these questions the

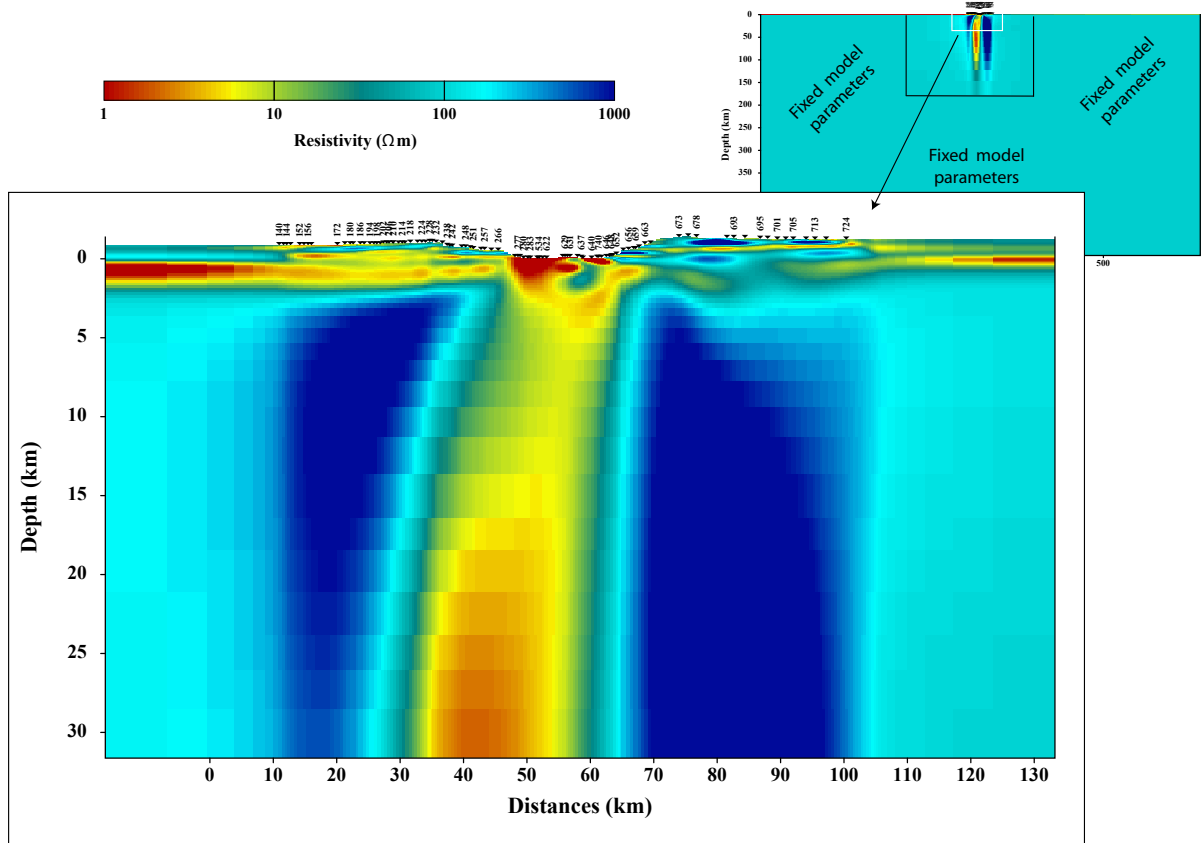


Figure 4.28: Inversion result after 1000 iterations with an RMS value of 1.83 using a constrained model with fixed resistivities ($100 \Omega m$) ~ 200 km to east and to the west from the profile and at a depth of ~ 200 km. The model shows that the inversion extends the western conductive layer at 2 km further to the west to replace the conductive layer which was originally placed at a depth of ~ 15 km to west from the profile. The upper 20 km of the model remain identical when compared with the structures of the model in Fig. 4.23b.

conductor is successively removed from the model, the forward modelling is used to test if any changes occur in the data fit. In this case one can assess if the conductor is required by the data. One could argue that the resolution tests discussed in Fig. 4.26 and 4.27 confirms that terminating the model at a certain depth worsens the data fit. However, the test in Fig. 4.27 cannot clarify if the poor data fit is caused because we removed the lateral conductivity contrasts west and east of the profile or because we terminated the central vertical conductor at depth.

Figure 4.30a shows a 2D model in which the central vertical conductor was cut by a resistive structure at a depth of ~ 70 km. Figure 4.30b shows the corresponding total RMS misfit at each station together with the values of the model presented in Fig. 4.28. It is clear from Fig. 4.30b that removing the central vertical conductor at a depth of ~ 70 km has a significant effect on the data fit. Particularly, effected are

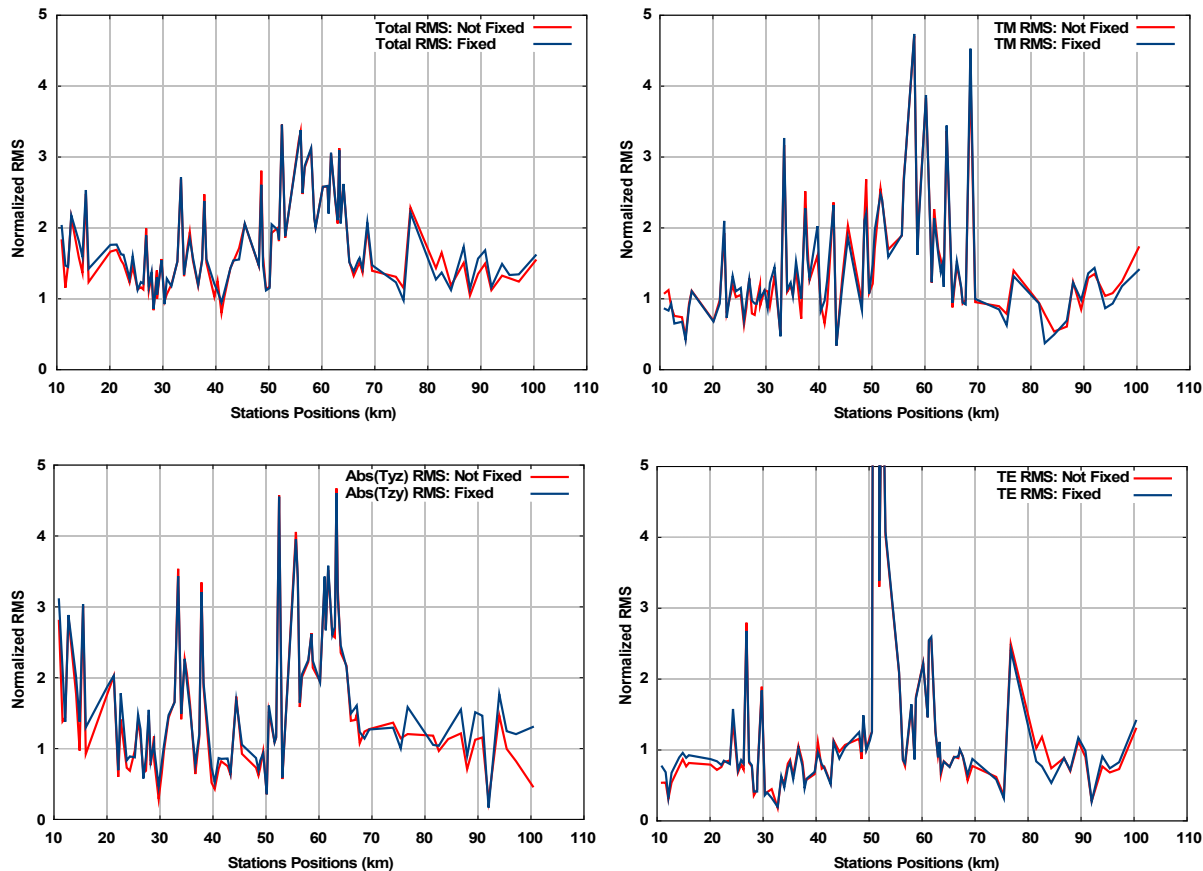


Figure 4.29: Comparison between RMS values computed for the model in Fig. 4.23b (red curves) and for the model in Fig. 4.28 (blue curves). This comparison reveals that there are only slight differences between the RMS values for the two models.

stations located at the eastern and western segments of the profile.

All RMS values of the central stations presented so far (i.e. Fig. 4.30b) show relatively high values. An explanation for this observation could be that these stations are more effected by 3D structures and off-profile features which cannot be modelled appropriately by the 2D inversion.

One possibility to improve the data fit at the central stations is to reduce the smoothing in the central part of the model, while keeping it unchanged for the rest of the model. Less smoothing will favour smaller structures, if they are required, to appear in the central part of the model to ensure a good data fit (see Fig. 4.16). However, care must be taken when reducing the smoothing in only one part of the model. Otherwise, discontinuities may appear between the central structures and the rest of the model.

Figure 4.31 shows the data misfits at three selected stations computed using the model in Fig. 4.28. The data misfit at station 278 shows a significantly high misfit partic-

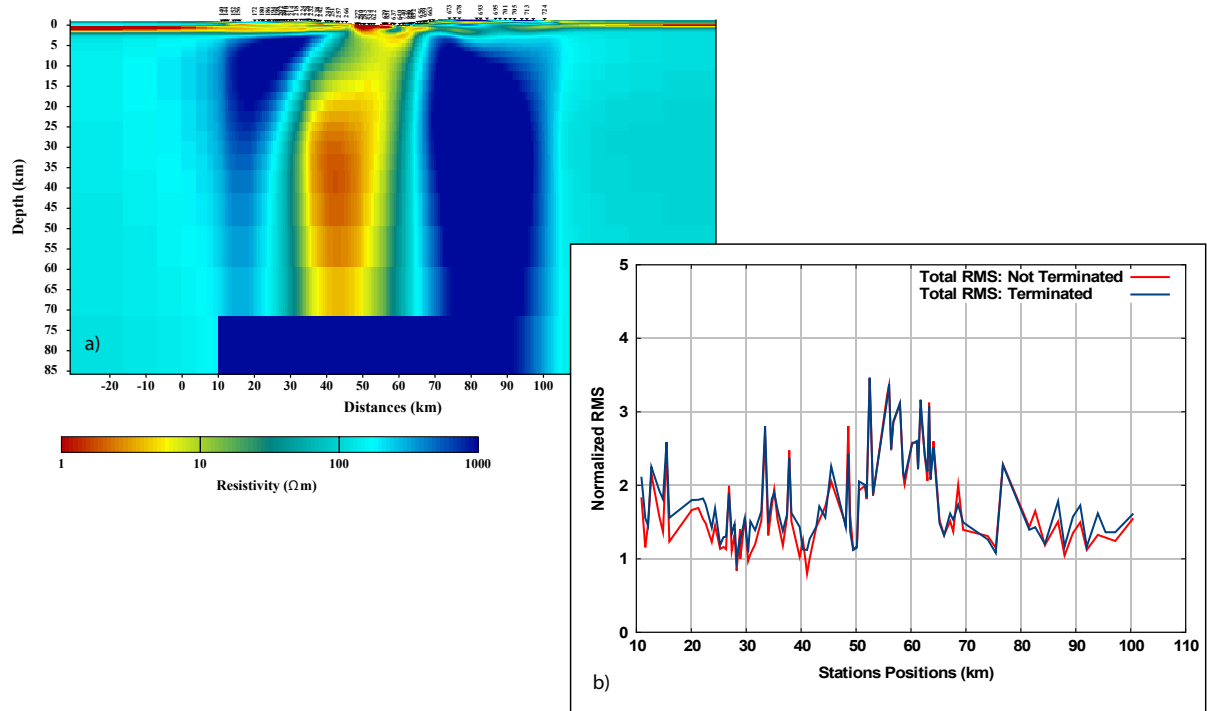


Figure 4.30: Forward modelling study to verify the presence of the central deep conductor and its depth extension. The total RMS values plotted for each station show that the conductor must extend to a depth of ~ 70 km to ensure a good data fit.

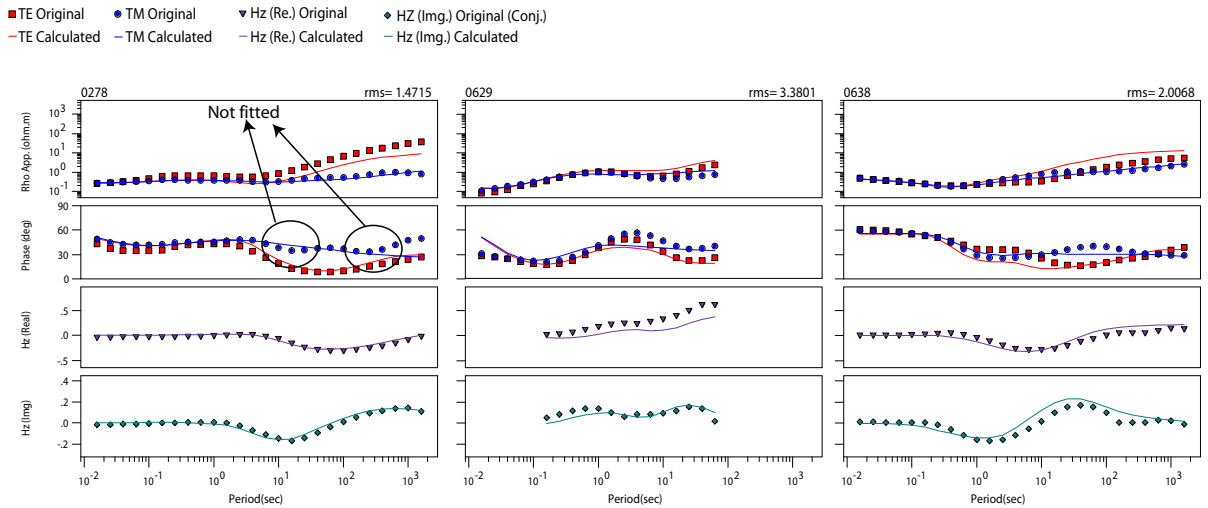


Figure 4.31: The data misfits at three selected stations to highlight the poor data fit at stations from the central segment of the profile. Particularly not fitted are the responses of small scale structures in the measured data, i.e. the ϕ_{TM} values at around 10s at station 278.

ularly for the ϕ_{TM} data⁴. Starting from ~ 2 s to the longest periods, the measured ϕ_{TM} curve show two period ranges (around 10s and 200s) for which the ϕ_{TM} values decrease before they increase at longer periods (> 300 s). The predicted ϕ_{TM} curve, however, decreases gradually from 2s onwards. This behaviour indicates that small scale structures in the sub-surface are required to fit the data. However, the applied smoothing for the inversion causes a smearing of small scale structures which appear to be connected with neighbouring structures. A similar behaviour, but in a different period range can be observed at stations 629 and 638.

To improve the data fit at the central stations, the model in Fig. 4.28 is used. However, the western and eastern parts of the model were kept fixed, while the central part of the model was free for changes during the inversion. The regularization parameter for the model in Fig 4.28 ($\tau=20$) was reduced to $\tau=10$. Figure 4.32b shows that after 200 iterations the global RMS value is reduced slightly from 1.8 to 1.7. A comparison between the two models in Fig. 4.32a and b and their computed total RMS values (Fig. 4.32c) reveals that changes in the model occurs mainly in the upper 5 km of the central part. For example, the relatively shallow resistive structure between profile km 55 and 60 is more pronounced in the model in Fig. 4.32b than in 4.32a. As the total RMS values in Fig. 4.32c indicate, the changes in the model improve slightly the data fit at the central stations.

The model presented in Fig. 4.33 focuses on the upper 20 km of the model. The structures presented in this model are the most robust features that the model must include to ensure a reasonable data fit for the 2D inversion. Furthermore, this model recovers all structures that have been found when inverting the individual components of the MT responses separately and jointly. These structures include the resistive blocks labeled A and B in Fig. 4.33, starting at a depth of ~ 2.5 km. In the western segment of the profile and up to a depth of ~ 5 km, a sequence of resistive-conductive layers extend from the central part of the model towards the west (label C). A very conductive structure ($< 4 \Omega m$) that is delimited laterally between profile km 45 and 65, extends to a depth of ~ 5 km (label D). The most conductive part of this structure ($< 1 \Omega m$) is located between profile km 45 and 50 and extends to a depth of ~ 1.5 km. The conductivity values decrease gradually with depth to reach values of $\sim 4 \Omega m$ at a depth of ~ 5 km. At the eastern part of this conductive feature (between profile km 55 and 60; label D1) relatively high resistive material ($> 100 \Omega m$) interrupts this conductive layer. The resistive intrusion extends laterally only few kilometres and has a limited depth of ~ 3 km.

Further to the east, similar conductive-resistive sequences can be observed beneath the eastern segment of the profile (label E). The first conductive layer appears at a depth of ~ 0.5 km (label E1). The thickness of this layer varies between 0.5 km

⁴Please keep in mind that high error floors were assigned to the ρ_{TE} data. Therefore, for clarity the error bars are not plotted.

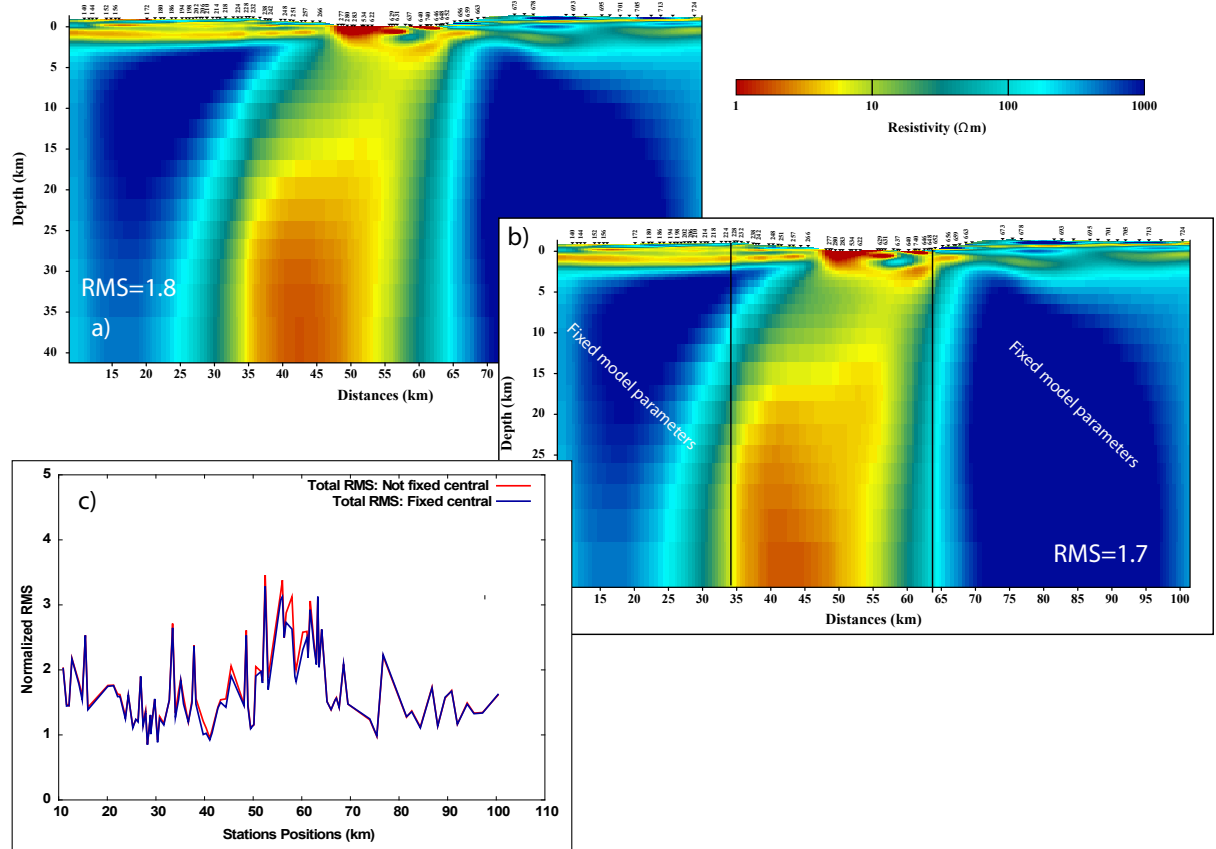


Figure 4.32: A comparison between a) the model presented in Fig. 4.28 with $\tau = 20$ and b) a model in which the areas to the east and to the west from the central segment have been fixed during the inversion and τ has been reduced to 10. A slight improvement in the global RMS can be observed. Furthermore, the shallow resistive feature in the central part appears more enhanced.

and 0.1 km. The underlying layer shows resistivity values $> 100 \Omega m$. The bottom conductive layer ($\sim 15 \Omega m$) in this sequence appears to be discontinuous along the eastern segment of the profile (label E2). This layer could reappear at shallower depth between profile km 75 and 80 smearing with the conductive structures in the central part of the profile.

The vertical conductor (F) in the central part of the profile is a dominating feature, which appears in all inverted models. In the model presented in Fig. 4.28, this conductor is located at a depth of ~ 16 km and extends down to ~ 80 km. The highest conductivity values ($< 3 \Omega m$) of this conductor occur at a depth of ~ 50 km. However, the smoothing of the inversion makes it difficult to determine exactly the position of the boundary between the vertical conductor and the two adjacent resistive blocks.

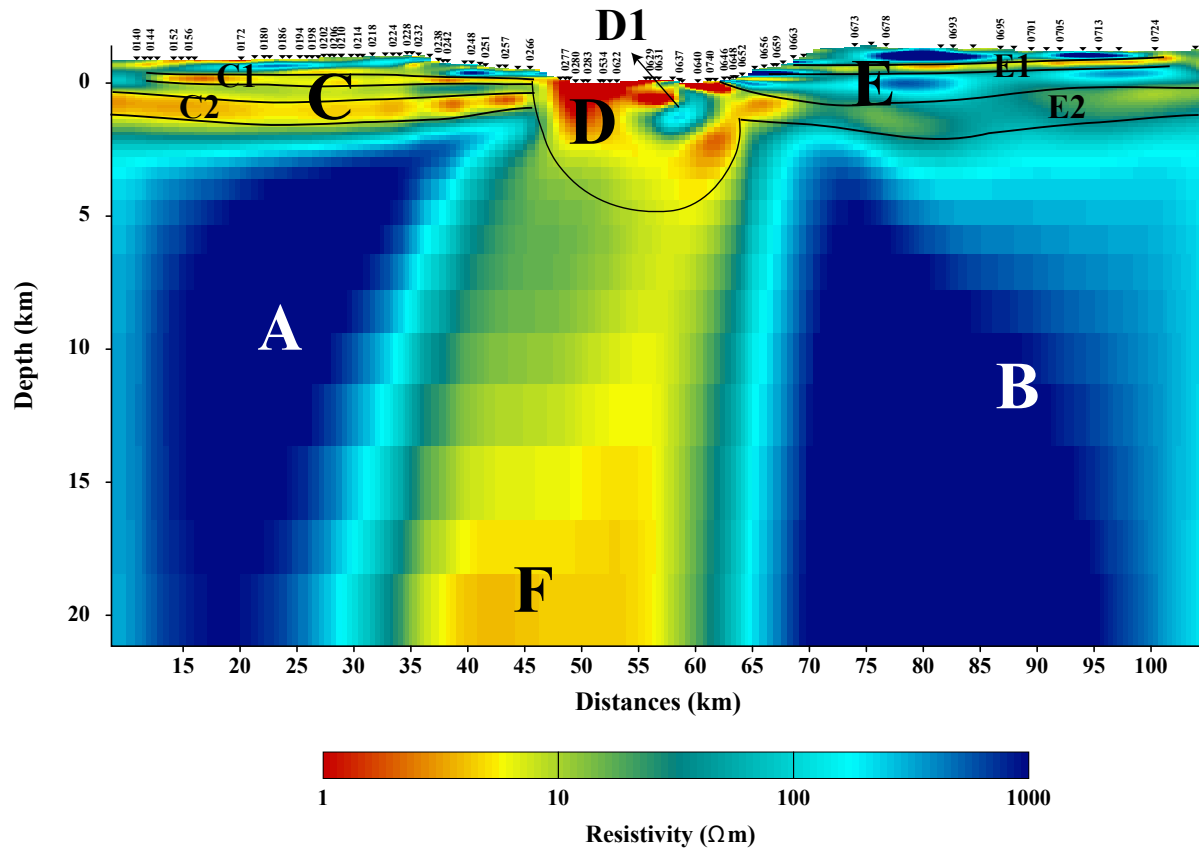


Figure 4.33: The preferred 2D conductivity model showing the most robust structures up to a depth of ~ 20 km. A and B are the two resistive blocks which start at a depth of ~ 2.5 km. They are separated by the central conductive part of the model F. C is the western resistive-conductive sequence which contains two conductive layers C1 and C2. D is an extremely conductive feature which spans the central segment of the model and extends to a depth of ~ 3 km. D is interrupted by the resistive block D1 which extends to a depth of ~ 2.5 km. E is the western resistive-conductive sequence which contains the shallow conductive layer E1 and a deeper, more discontinuous layer E2.

4.5.4 Interpretation of the conductivity model

So far, the discussion was focused on conductivities and their distribution in the model to ensure the best fit for the various MT data components. But to which degree does this conductivity model reflect the geological and tectonics setting of the study area? A comparison with the available geological cross-sections and stratigraphic columns from deep boreholes helps to interpret the conductivity distribution of our 2D model from geological and tectonic points of view.

Based on surface geology, boreholes information, gravity and magnetic measurements, a geological cross-section in the vicinity of the study area (Fig. 4.34b) was published by Garfunkel & Ben-Avraham (1996). The geological cross-section extends approximately between profile km 35 and 75 and summarise the main geological formations and the tectonic setting of the upper 10 km (Fig. 4.34a).

A comparison between the conductivity model and the geological cross-section in Fig. 4.34a reveals:

- The location of the relative high resistive feature (D1 in Fig. 4.33) which is embedded in the central shallow conductive structures coincides with the location of the Al-Lisan salt diapir. Based on the stratigraphic column obtained from the Al-Lisan1 deep borehole (see Fig. 4.34b for location), the Al-Lisan salt diapir consists mainly of Halite (Powell, 1988). The salt diapir appears in the stratigraphic column of the Al-Lisan1 deep borehole at a depth of ~ 200 m and extends to ~ 3.5 km (the total depth of the Al-Lisan1 borehole). In our conductivity model, resistive feature labeled D1 appears at a depth of ~ 150 m and extends to ~ 2.5 km. The relatively high resistivity values of the Halite is an indication that it is dry e.g. (Telford & Sheriff, 1990).
- The conductive structure D that appears in the central part of the profile and extends to a depth of ~ 3 km, coincides with the recent sediments of the Quaternary (the Al-Lisan formation) which consist mainly of gravel with an intercalation of salt and clay sediments (Powell, 1988). The extremely high conductivity values ($< 1 \Omega m$) in this part of the model indicate that the Dead Sea brines must reach depths of several kilometres.
- The lateral transition from the conductive structures in the central part to the eastern resistive block B in Fig. 4.33 which starts at a depth of ~ 2 km coincides with trace of the left lateral strike-slip Eastern Boundary Fault (EBF). The transition from the Cretaceous and the Jurassic sediment formations to the crystalline Precambrian igneous rocks at the eastern part appears at a depth of ~ 1 km in the geological cross-section. However, in the conductivity model a vertical transition from the eastern resistive-conductive sequence layers (E in Fig. 4.33) to the resistive eastern block B is imaged at a depth of ~ 2.5 km.

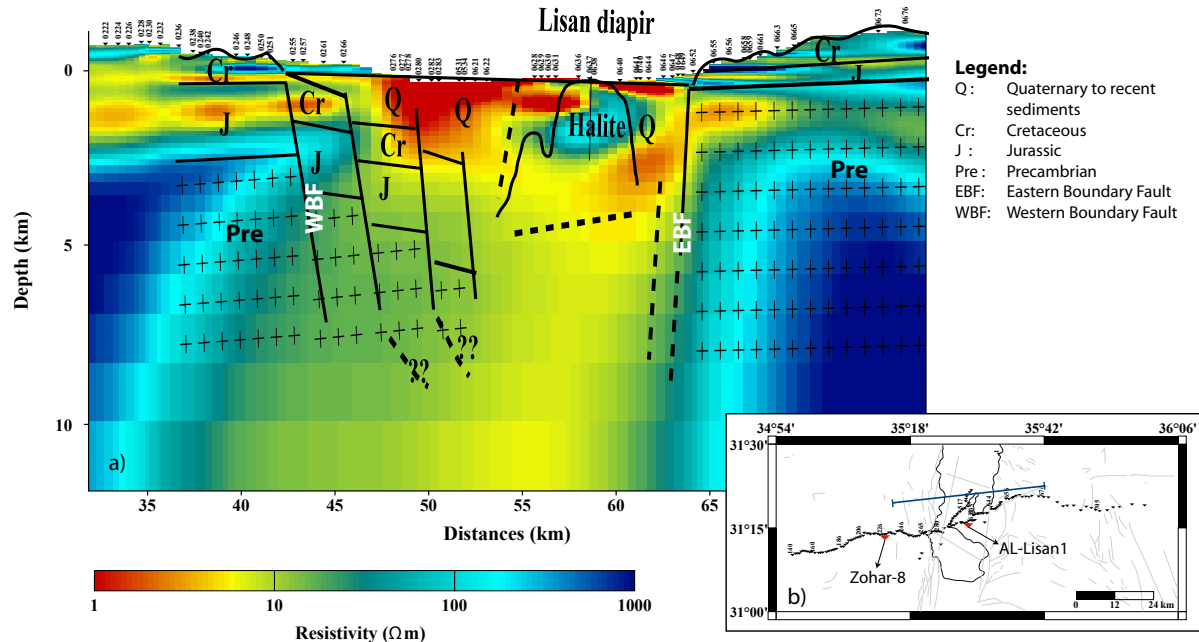


Figure 4.34: Comparison between the preferred conductivity model (Fig. fig:final-20km) and a geological cross-section modified after Garfunkel & Ben-Avraham (1996), see also Fig. 4.33. The location of the Halite intrusion defined in the geological cross-section coincides with feature D1. The lateral transition from the western resistive block B to the central deep conductor F coincides with the trace of the Eastern Boundary Fault (EBF). However, the top of the Precambrian basement (Pre) does not coincide with the top of the block B. The location and the depth extension of the Quaternary formation (Q) coincides with the conductive feature D. At the western side, the interface between the Cretaceous (Cr) and the Jurassic (J) is located at the bottom of the shallow conductive layer C1. The thick Jurassic formations contain in its middle part the deep conductive layer C2. The interface between the upper crustal sedimentary formations (Cretaceous and Jurassic) and the crystalline Precambrian igneous rocks is located at the top of the western resistive block A.

- The lateral transition from the conductive structures in the central part (F) to the western resistive block A in Fig. 4.33 appears more smoothly and not as a sharp conductivity contrast at the eastern part. Therefore, it is difficult to identify the Western Boundary Fault (WBF). Nevertheless, in the conductivity model at a depth of ~ 2 km, there is a clear transition from the central upper conductor D and the western resistive block A.
- The transition from the Cretaceous and the Jurassic sediment formations to the crystalline Precambrian igneous rocks in the western part seems to coincide with the transition between the western resistive-conductive sequence C to the resistive western block A. Moreover, the transition from the Cretaceous

to the Jurassic formations could be reflected in the transition from the upper conductive-resistive sequence C1 to the lower conductive-resistive sequence C2.

Additional geological information are available from the stratigraphic column of the Zohar-8 deep borehole (Kashai & Croker, 1987). This deep borehole is located a few kilometres to the south of the western segment (see Fig. 4.34b for location) in the Zohar oil field. The Zohar-8 deep borehole was drilled mainly for gas and oil production with a total depth of ~ 3 km. The stratigraphic column of the Zohar-8 deep borehole shows more details concerning the geological structures at the western side of the study area. A comparison between the stratigraphic column of the Zohar-8 deep borehole and the conductivity model is shown in Fig. 4.35.

The transition from the Cretaceous to the Jurassic seems to be marked by a transition from the upper to the lower resistive-conductive sequence and coincides with lower boundary of the conductive layer C1. Moreover, the lower Cretaceous Kurnub group may be linked with conductive layer C1. The resistive layer in the lower resistive-conductive sequence seems to coincide with the Zohar and Sherif groups which are part of the upper Jurassic. The transition from the Jurassic to the Triassic is not expressed in the conductivity. Instead, the lower conductive layer C2 appear to extend vertically from the middle Jurassic to the middle Triassic with a total thickness of ~ 1.5 km. The Permian is located somewhere at the transition where the smoothing of the inversion smears the structures between the conductive structures C and the resistive block A. However, the transition from the upper sedimentary formations including the Cretaceous, Jurassic, Triassic and Permian to the crystalline Precambrian igneous rocks is clearly identified as a transition from resistive-conductive sequences C to the resistive block A in the conductivity model.

The conductive layer C1 in Fig. 4.33 could be interpreted as a shallow aquifer, especially when taking into account that the regional aquifers in this area are the Palaeozoic to lower Cretaceous Kurnub group aquifer and the upper Cretaceous Judea Group aquifer (Kafri *et al.*, 2008).

To the east of the Dead Sea in Jordan, the available information that can be used in terms of a geological interpretation of the conductivity model are surface geology, shallow boreholes and the Al-Lisan1 deep borehole. The stratigraphic column from Al-Lisan1 deep borehole indicates the salt body from a depth of ~ 200 m to at least 3 km, which coincides, as previously discussed, with the resistive feature D1. Surface geology and hydro-geological information to the east of the Dead Sea (Khalil, 1992) indicates the presence of the upper Cretaceous formation (Balqa/Judea⁵) and the lower Cretaceous (Kurnub sand stone formation). However, the Jurassic strata is missing in the stratigraphic column of the study area in Jordan. The Kurnub sandstone to the east of the Dead Sea is the main aquifer in Jordan. The underlying strata

⁵The upper Cretaceous formation is denoted as Balqa by geologists in Jordan and as Judea by geologists in Israel

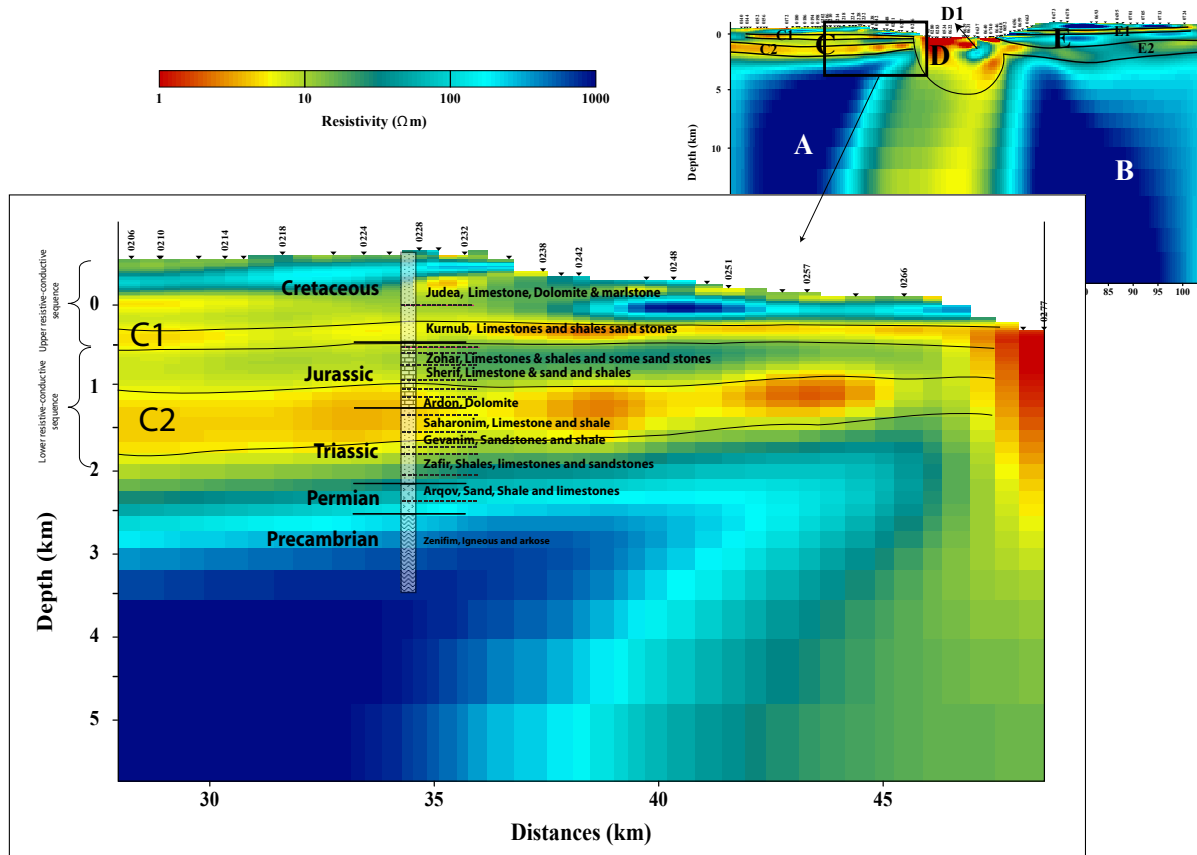


Figure 4.35: Comparison between the conductivity model and the stratigraphic column of the Zohar-8 deep borehole (Kashai & Croker, 1987). Two main interfaces can be identified in the conductivity model: The transition from the Cretaceous to the Jurassic and the transition from the upper crustal sedimentary formations (Cretaceous, Jurassic, Triassic and Permian) to the crystalline Precambrian igneous rocks. Moreover, the lower Cretaceous (Kurnub formation) coincides with the upper conductive layer C1. The gas and oil production formations in this borehole are the upper Jurassic formations (Zohar and Sherif) which coincide with the resistive layer sandwiched between C1 and C2. The deeper conductive layer C2 spans vertically from the middle Jurassic to the middle Triassic with a total thickness of ~ 1.5 km.

(Triassic-Permian) consists mainly of sandstone alternated with siltstone (Umm Irna sandstone). The Umm Ishrin sandstone (Cambrian) underlays the Umm Irna sandstone. The thickness of the Umm Ishrin sandstone varies in the study area from 300 to 330m, where only the upper 150-200m of the formation crop out (Powell, 1988; Khalil, 1992).

Taking into account these geological information from the eastern side of the study area, the shallow conductive layer E1 in Fig. 4.33 may coincide with the Kurnub sandstone of the lower Cretaceous. This hypothesis is supported by the fact that the depth to the top of E1 is comparable⁶ with the depth to the top of C1 which is correlated with the Kurnub formation in the stratigraphic column of the Zohar-8 borehole (Fig. 4.35). The deep conductive layer E2 maybe part of the Cambrian Um Ishrin sandstone which is defined as an aquifer (Khalil, 1992). The interface between the upper crustal formations (Cretaceous, Triassic, Permian and Cambrian) and the mid-crustal basement (Precambrian) is interpreted to coincide with the top of the resistive block B.

As pointed out in the introduction of this chapter, wide- and narrow angle refraction/reflection (WRR and NRR) seismic measurements were carried out in the frame work of the DESIRE project. The WRR measurements comprised 11 shots along a 235 km long profile which coincides with the MT profile (Fig. 4.36c). Recently, a 2D P-wave velocity model for the DESIRE WRR data was published (Mechie *et al.*, 2009). The 2D velocity model covers the upper 40 km of the earth's crust and mantle (Fig. 4.36a). A comparison between the 2D velocity model and the conductivity model (Fig. 4.36) for upper crustal to mid-crustal depth (~ 20 km) reveals:

- The top of the seismic basement, which is defined as the top of the layer with P-wave velocities greater than 5.6 km s^{-1} , varies in depths from ~ 2 km in the vicinity of the DSB to ~ 6 km at the eastern end of the seismic profile. Whereas, the top of the seismic basement within the DSB is located at a depth of ~ 10.5 km (Mechie *et al.*, 2009). In comparison to the conductivity model, the top of the seismic basement to the east and to the west of the DSB agree with the tops of the western and the eastern resistive blocks A and B which appear at depths of ~ 2.5 km. However, within the DSB there is no correlation between the velocity and conductivity models concerning the top of the basement.
- The sharp lateral contrast in the conductivity between the central part of the model (F) and the adjacent resistive blocks A and B, agrees to a depth of ~ 8 km with the sharp lateral contrast from low velocities for the in-basin sedimentary fill ($< 4 \text{ km s}^{-1}$) and the adjacent seismic basement.

⁶Considering the topographic differences between the eastern and the western segments of the profile

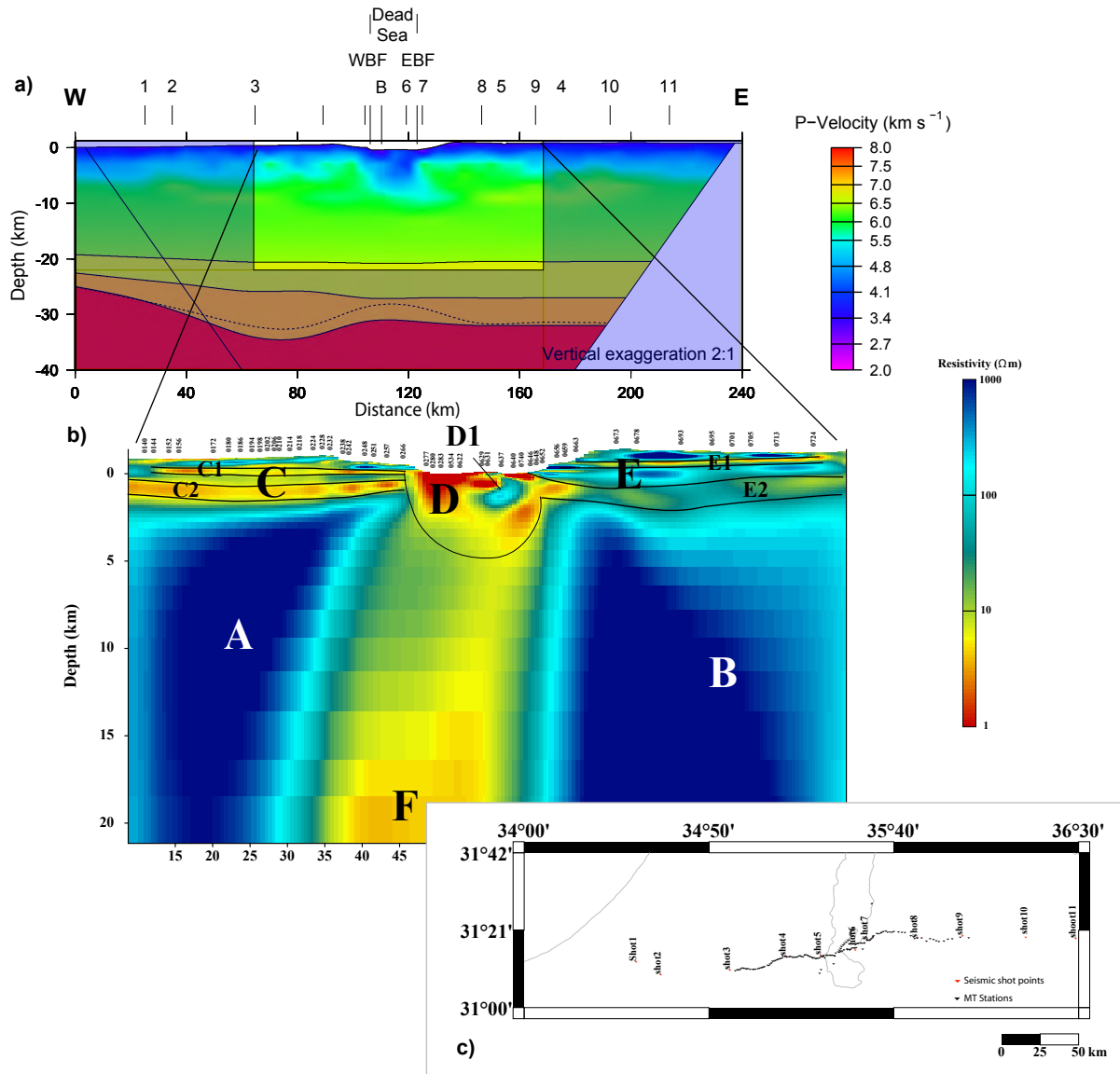


Figure 4.36: Comparison between a) the 2D P-wave velocity model after *Mechie et al.* (2009) and b) the conductivity model obtained in this study. c) A map showing the locations of the seismic shot points relative to the MT stations. The top of the seismic basement (green colour; P-wave velocities $> 5.6 \text{ km s}^{-1}$) to the west and to the east of the Dead Sea coincides with the top of the resistive blocks A and B, respectively. Down to a depth $\sim 8 \text{ km}$, the sharp conductivity contrasts between F and A, B has a similar expression in the seismic model as a sharp velocity contrast. The upper crustal formations (up to 5 km) are not well expressed in the seismic model.

- The Upper 3 km cannot be compared, since the wide-angle velocity 2D model does not reveal any structures in this depth range.

4.5.5 Three dimensional modelling and inversion of DESIRE magnetotelluric data

The geo-electrical strike direction and dimensionality analyses discussed in section 4.4 revealed that 3D effects are present in the data, in particular at stations located in the central part of the study area (Dead Sea area). Even without these analyses, it is quite obvious that the shape of the Dead Sea and the underlying sedimentary basin represent a strong 3D feature which has an influence in the data. Therefore, we collected the DESIRE-MT data along a main profile and a shorter profile (in-basin profile; Fig. 4.5) to achieve some 3D areal coverage. However, two main points must be considered before discussing the results from the 3D inversion: i) The set-up used for the DESIRE-MT station provides only to some extent spatial station coverage, which means that this set-up is not ideal for a 3D interpretation. The majority of stations are distributed along the main profile. Only in the central part (in the basin), where the main and in-basin profiles cross each other there is a good station coverage. ii) The Modular system code used for the 3D inversion still under development stage. So far, the 3D inversion tests of the Modular system, only synthetic data were used to test the accuracy and generality of the code. In my work, it is the first time that the code is used to invert real MT data in 3D. Moreover, several parameters control the inversion run. The influence of these parameters on the inversion results has not been properly tested yet. Therefore, care must be taken when interpreting the 3D inversion results.

To invert the DESIRE-MT data, the parallel 3D inversion scheme that I have developed in the framework of this thesis was used. I would like to start this topic with some 3D forward modelling to explain certain aspects of the measured data. For this 3D forward modelling, only the vertical transfer functions in terms of induction vectors are analysed, because this data type is very sensitive to lateral conductivity contrasts.

Three dimension modelling of the DESIRE-MT data

Starting with a 3D model which contains only structures that are visible at surface, a regional 3D model was constructed. The regional 3D model extends laterally from -1000 km to 1000 km in east-west and north-south directions and to a depth of ~ 300 km (Fig. 4.37a). The 3D model consists of conductive structures ($0.3 \Omega m$) simulating the Mediterranean Sea in the north west direction and extending to a depth of ~ 5 km (Fig. 4.37a). Another conductive structure ($0.3 \Omega m$) which extends to a depth of ~ 1.5 km is included to simulate the Red Sea (Fig. 4.37a). The central part of

the 3D model contains two conductive structures ($0.1 \Omega m$ and $1 \Omega m$; Fig. 4.37b) to simulate the northern and southern Dead Sea brines. The structures which simulate the Dead Sea extend to a depth of ~ 1.5 km. The background was set to a resistivity of $100 \Omega m$. In total, the 3D model consists of $109 \times 94 \times 37$ grid cells in X, Y and Z directions, respectively.

For the 3D forward modelling, 25 periods covering the period range from $10^{-2} s$ to

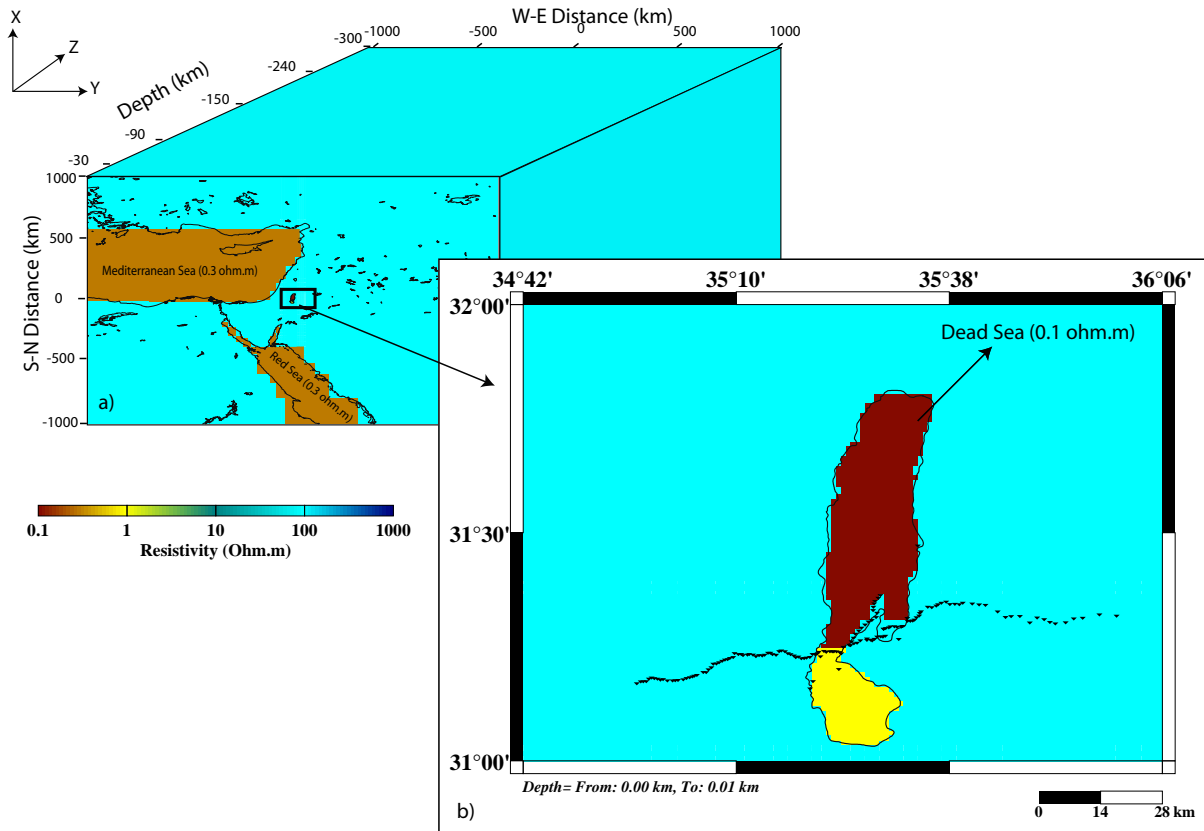


Figure 4.37: A regional 3D model used to analyse the behaviour of the measured induction vectors. a) The model contains conductive structures ($0.3 \Omega m$) simulating the Mediterranean Sea in the NW direction and the Red Sea in the south. b) The central part of the model contains two conductive structures (0.1 and $1 \Omega m$) simulating the northern and southern part of the Dead Sea. The Mediterranean Sea structures extend to a depth of ~ 5 km, whereas the Dead Sea and the Red Sea extend to a depth of ~ 1.5 km.

$\sim 3000s$ and all MT stations located at the main and in-basin profile were used. For the parallel scheme, 26 processors were used, one for each period and a master. To solve the forward modelling problem on the parallel machine took approximately 147 minutes. Whereas, solving the same forward modelling problem in the normal serial way took approximately 2220 minutes (1.5 day). This means that the 3D forward modelling speed up factor for this particular model is approximately 15.

Since the Mediterranean Sea and the Dead Sea conductive structures represent large lateral conductivity contrasts, the induction vectors are very sensitive to these in the measured data. Figure 4.38 shows the real parts of the predicted (modelled) induction vectors (black vectors) together with the measured data (red vectors) at five selected periods.

Figure 4.38 indicates that at short periods (1s), the predicted induction vectors at stations of the Al-Lisan peninsula and to the west and east of the Dead Sea start to point away from the conductive structures in the central part of the model. One can clearly see a discrepancy in the amplitude between the measured and calculated induction vectors: While the amplitude of the predicted induction vectors reaches 1 and above at some stations located at the Al-Lisan peninsula, the amplitude of the measured induction vectors remains very small. This indicates that the conductivity contrast of the 3D model between the saline water of the Dead Sea ($0.1 \Omega m$) and the Al-Lisan peninsula ($100 \Omega m$; Fig. 4.37) is too high. It is therefore more likely that the shallow structure of the Al-Lisan peninsula is much more conductive ($< 100 \Omega m$). As shown in Fig. 4.9, the influence of conductive structures in the central part of study area starts to appear in the measured data at intermediate periods (32s in Fig. 4.9 and 45s in Fig. 4.38). This influence is confirmed by the modelling because the predicted induction vectors in the same period range show a similar behaviour. An interesting result can be observed when comparing the directions of the measured and predicted induction vectors at 45s: Stations at the western shore of the Dead Sea show a good agreement between the measured and predicted induction vectors in both amplitude and direction. However, stations located at the Al-Lisan peninsula and at the eastern coast of the Dead Sea show noticeable differences in the directions of the induction vectors. The predicted induction vectors indicate N-S extending conductive structure of the Dead Sea, but the measured induction vectors have much more pronounced southward directions. This could be related to a conductor located to the north of the eastern segment of the main profile. Furthermore, the amplitudes of the induction vectors at stations located over Al-Lisan peninsula could indicate a higher lateral conductivity contrast at depth between the Dead Sea and the structures beneath the peninsula.

This kind of behaviour of the induction vectors can be observed to periods up to 360s. However, the amplitudes of the predicted induction vectors decrease gradually, while the amplitudes of the measured induction vectors remain high. This indicates that the central conductivity contrast between the Dead Sea and the Al-Lisan peninsula structures must extend deeper to ensure the high amplitudes of the measured induction vectors.

At periods longer than 700s, the influence of the Mediterranean Sea becomes noticeable in the predicted data, most obviously at the western stations. In comparison to the induction vectors computed with the 2D forward modelling code (Fig. 4.15),

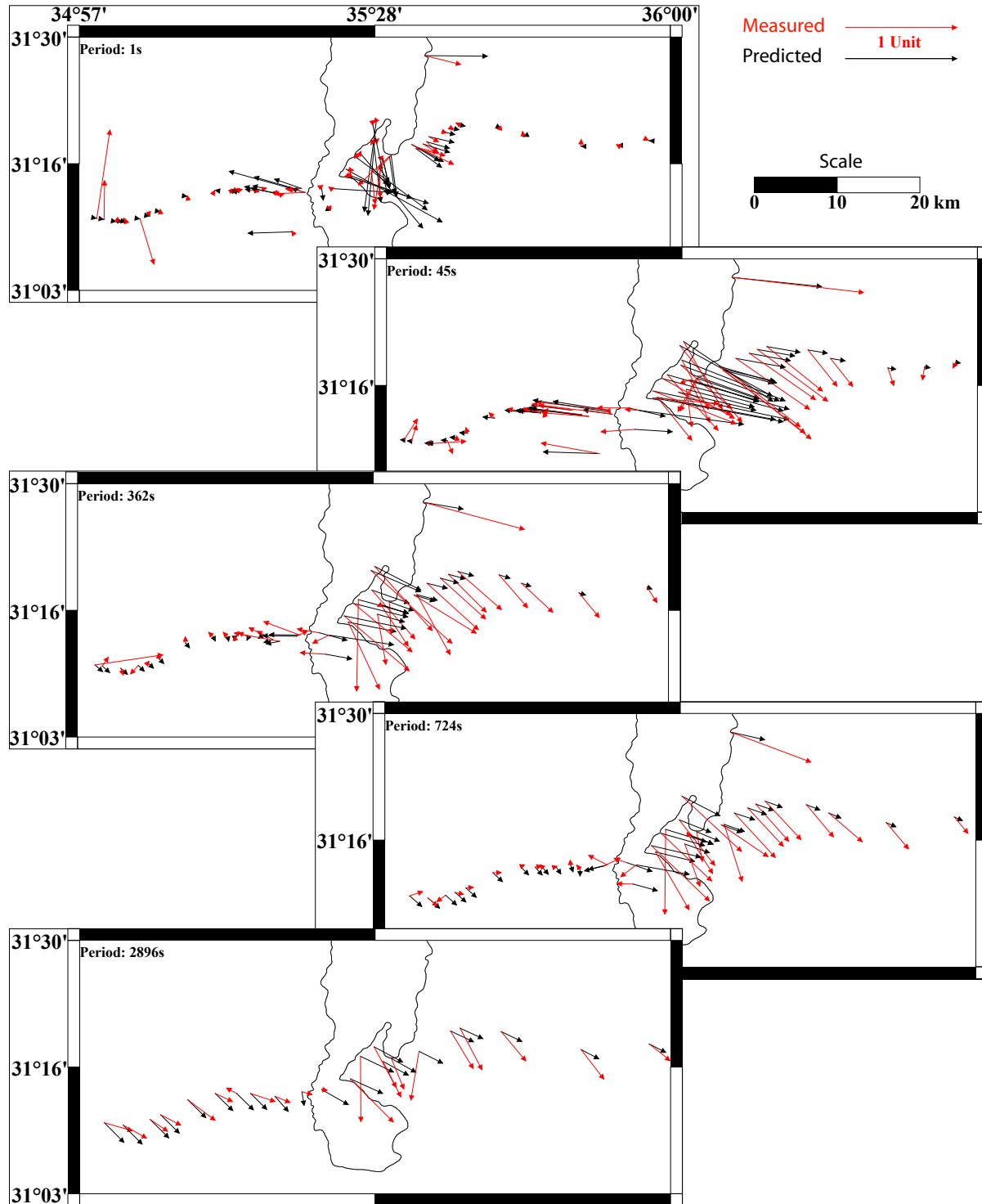


Figure 4.38: The real parts of the predicted induction vectors (black vectors) computed for the regional 3D model of Fig. 4.37, together with the measured induction vectors (red vectors). Due to the high conductivity contrast between the Al-Lisan structures ($100 \Omega m$) and the Dead Sea brines ($0.1 \Omega m$) in the NW, the predicted induction vectors show high amplitudes even at shorter periods (1s). Almost all stations are influenced by the conductor in the central part at a period of 45s. This influence decreases at periods longer than 362s and the influence of the Mediterranean Sea starts to appear at the western stations.

the induction vectors in Fig. 4.38 show a stronger tendency to point southwards. This is because the Mediterranean Sea coast line does not run in N-S direction, but in NE-SW direction (Fig. 4.37). At the longest periods (2889s), the amplitudes of the measured and predicted induction vectors show good agreement, particularly at the western stations. However, the measured induction vectors point more eastwards than the predicted. A deep conductor that is located to the south of the western segment of the main profile may be necessary to explain this behaviour.

Considering these arguments, a new 3D model was constructed (Fig. 4.39). The new 3D model contains the same regional structures (Mediterranean Sea and Red Sea), but the central part was modified. The conductive structures of the northern and the southern parts of the Dead Sea are extended to a depth of ~ 3.5 km. In the shallow part (up to 700 m), the Al-Lisan peninsula is made more conductive ($0.5 \Omega m$). From 700 m down to ~ 3.5 km, resistive structures are embedded beneath the Al-Lisan peninsula. 15 km to the north of the eastern segment of the main profile, a new conductive layer ($1 \Omega m$) is introduced. It starts at 1.1 km and extends to a depth of 6.7 km.

The same period layout as before was used. Figure 4.40 shows the real parts of the predicted induction vectors (black vectors) for the model presented in (Fig. 4.39) together with the real parts of the measured induction vectors (red vectors).

Figure 4.40 shows that the inclusion of the new structures generally improves the data fit. For the Al-Lisan peninsula, the amplitudes of the predicted induction vectors at short periods (1s) are now smaller and compatible with the measured data. The influence of the conductor to the north of the eastern segment of the main profile appears in the predicted data at periods longer than 1s and reaches its maximum at 45s. Now, the predicted induction vectors point more southwards. The inclusion of the high resistive structure beneath the Al-Lisan peninsula at a depth greater than 700 m, forces the predicted induction vectors at stations located on the Al-Lisan peninsula to point more to the south. This behaviour can be observed at periods longer than 10s. Again, the influence of the Mediterranean Sea at the western stations can be observed at periods longer than 700s. The predicted induction vectors at the western stations are very similar to the previous result in Fig. 4.40, since no additional structures were added to the west of the Dead Sea. At the longest periods (2896s), the predicted induction vectors at the eastern stations start to point more to the east indicating that the influence of the northern conductor starts to disappear. However, the measured induction vectors continue to point more to the south. This means that the new introduced northern conductor must extend deeper than 6.7 km.

The modelling of the induction vectors can be summarised as follows:

- The shallow structures of the Al-Lisan peninsula must be conductive to ensure low amplitudes of the induction vectors for periods less than 1s. At greater depth, resistive structures are required to reproduce the amplitudes and direc-

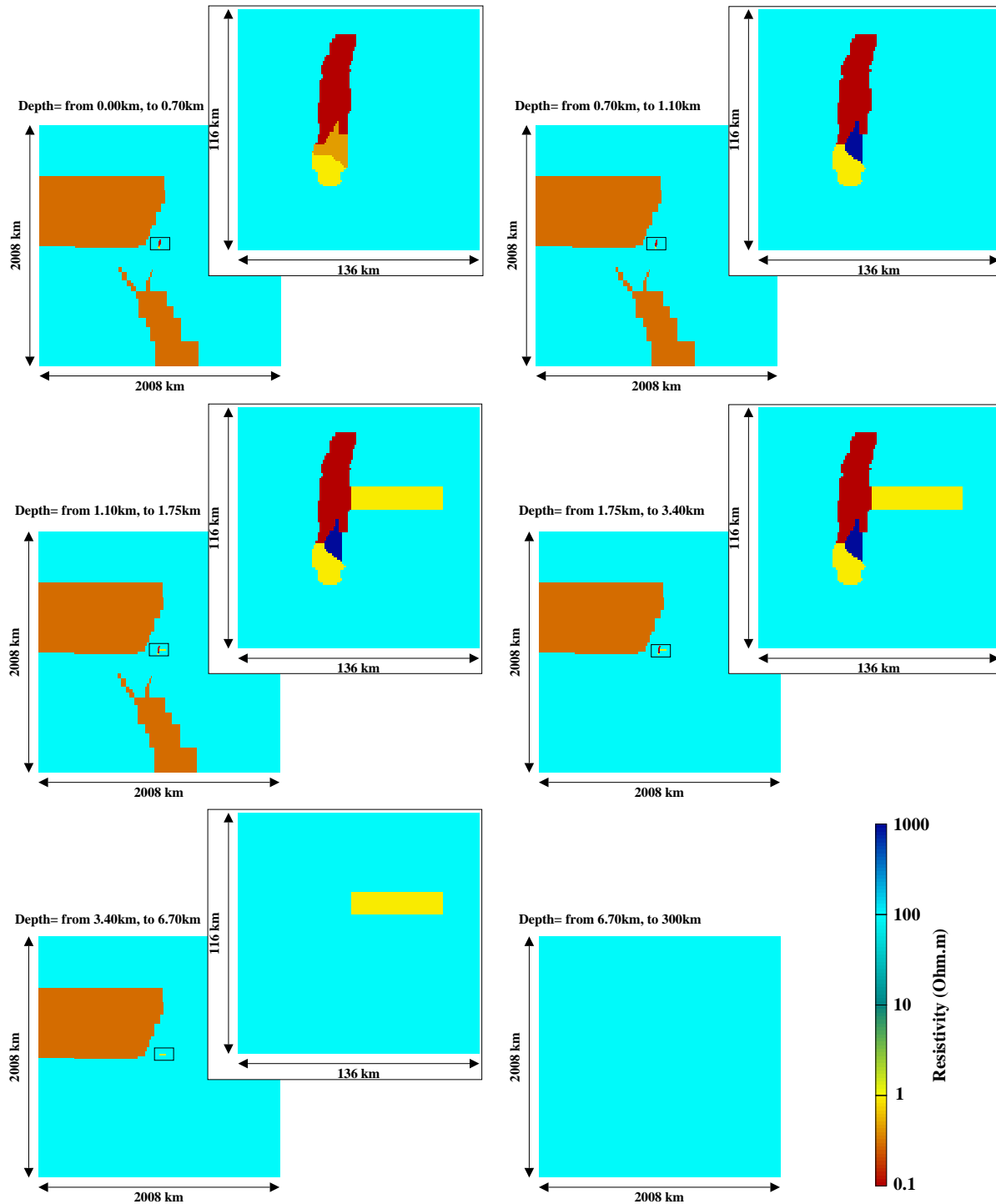


Figure 4.39: A modified version of the 3D model presented in Fig. 4.37. The resistivity of the shallow structures (up to 700 m) of the Al-Lisan peninsula is reduced from $100 \Omega m$ to $0.5 \Omega m$. From a depth of 700 m to 3.4 km, a resistive feature ($1000 \Omega m$) which takes the form of the Al-Lisan peninsula is included in the model. 15 km to the north of the eastern segment of the profile, a conductive layer ($1 \Omega m$) is included which extends from a depth of 1.1 km to 6.7 km.

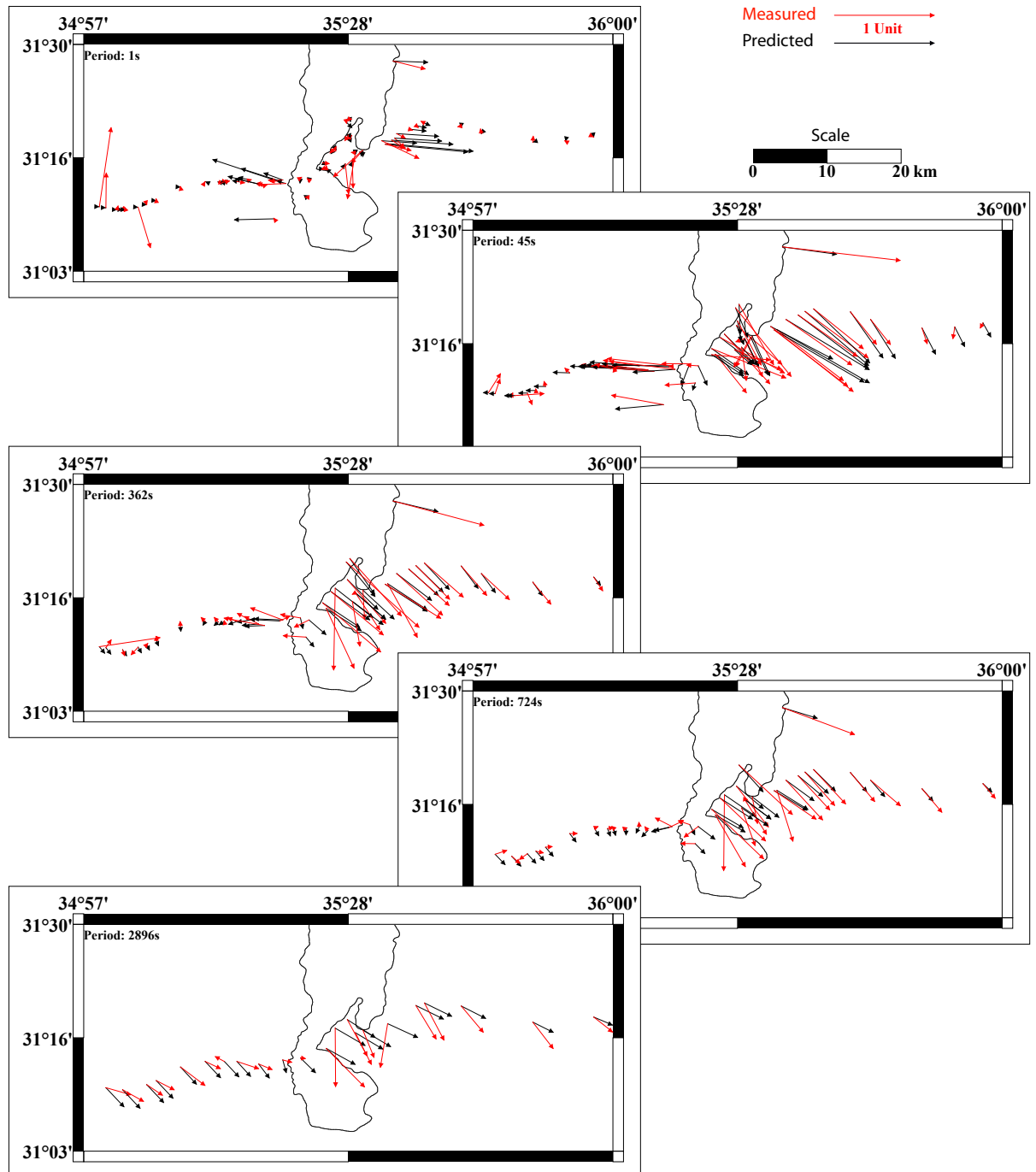


Figure 4.40: The real parts of the measured (red vectors) and predicted (black vectors) induction vectors for the modified model in Fig. 4.39. Reducing the resistivity of the shallow structures of the Al-Lisan peninsula from $100 \Omega m$ to $0.5 \Omega m$ ensures low amplitudes of the predicted induction vectors at short periods. The inclusion of the new conductive layer towards the NE and increasing the resistivity of the deep structures of the Al-Lisan peninsula make the predicted induction vectors more compatible with the measured data at periods from 45s to 724s.

tions of the measured induction vectors of the Al-Lisan peninsula stations.

- The tendency of the measured induction vectors at the eastern stations to point in south-east direction requires an additional conductor to the north of the eastern segment of the study area in addition to the Dead Sea brines.
- The conductive structures of the Mediterranean Sea have an impact at periods longer than 700s in particular at the western stations. Furthermore, the tendency of the measured induction vectors at the western stations to point in south east direction could be a result from the NE-SW trending direction of the Mediterranean coast line.
- The conductive structures of the Red Sea in the south, appear to have no influence on the data.

Three dimensional inversion of the DESIRE-MT data

I used the Modular system, written by G. Egbert and A. Kelbert (personal communication), and parallelized in the scope of my thesis, to invert the DESIRE-MT data in 3D.

As pointed out previously, inverting the MT data first separately and then jointly provides a better idea which structures are associated with each data type. For the 3D inversion, however, I could not successfully invert the DESIRE-MT data jointly. This is due to difficulties in defining a proper error level for the impedance tensor components and the vertical magnetic transfer functions. For this reason, only results of fitting the vertical magnetic transfer functions data and the impedance tensor data separately are discussed.

I started the 3D inversion by inverting the vertical magnetic transfer functions T_{zx} and T_{zy} . Data from 96 stations located at the main and in-basin profiles and 20 periods covering the period range from 0.001s to 3000s were used. The 3D model contains $94 \times 87 \times 37$ model parameters in X, Y and Z directions, respectively. Additionally, 10 air layers were added in Z-direction. The 3D model extends from -1000 km to 1000 km in E-W and N-S directions and to a depth of ~ 300 km. A homogeneous half space of $50 \Omega m$ was used as a starting model. I did not use the synthetic model presented in Fig. 4.39 to obtain only data-driven 3D inversion model.

For the parallel scheme, 21 processors (one processor for each period and one as a master) were used. It is worthwhile to notice that the NLCG inversion scheme (Non linear conjugate gradient, see sec. 2.6) does not require an explicit use of the sensitivity matrix. Therefore, the parallelization is applied over periods for solving the forward modelling problem as well as for the 3D inversion.

Figure 4.41 shows the 3D inversion result after 108 iterations with a global RMS value of 4.6. Note that the computed global RMS value is subject to the added error floors and model covariances which are still under test. The result is presented as a plane view at different depths.

The 3D inversion result shows that most of the structures from the forward modelling study (Fig. 4.39) reappear. The most obvious conductive structure in the model is the feature labeled A. This feature starts to appear at a depth of ~ 200 m, disappears at a depth of ~ 20 km and has a NE-SW extension. The resistivity values of this feature decrease gradually from $\sim 10 \Omega m$ at ~ 300 m depth to reach a minimum value of $0.2 \Omega m$ at ~ 2.2 km depth. From a depth of ~ 2.2 km down to ~ 20 km, the resistivity values increase gradually and the feature A smears with the less conductive underlying structure. At a depth of ~ 3.5 km, feature A starts to concentrate mainly at the northern part. The resistive zone labeled B is also pronounced in the model and is embedded within the conductive feature A. This zone appears at a depth of ~ 900 m and disappears at ~ 20 km depth. The resistivity values of the structures in zone B increase from $\sim 300 \Omega m$ at surface to reach the maximum at a depth range $\sim 2-4$ km

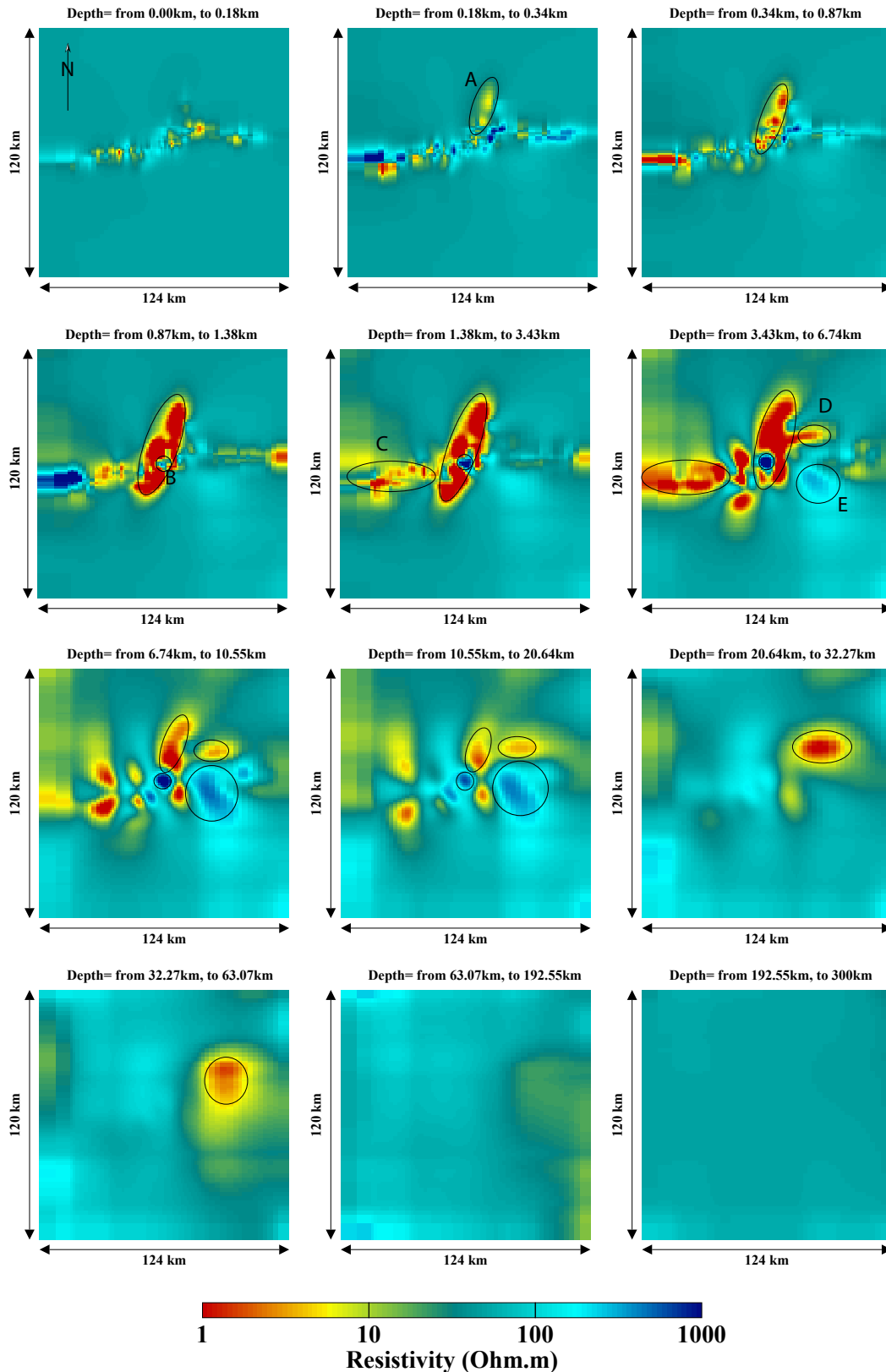


Figure 4.41: The 3D inversion result fitting only the T_{zx} and T_{zy} data. The 3D model is presented as slices in N-S, E-W directions at different depths. The most pronounced structures are labeled as A, B, C, D and E according to their first appearance in the model. In total, the model contains 37 depth slices, however only 9 slices are presented. See text for explanation.

and increase at greater depths. Another conductive structure which is pronounced in the model with an E-W extension is the feature labeled C. This feature appears at a depth of ~ 1.4 km. The resistivity values of this feature vary between $1 \Omega m - 10 \Omega m$, whereas minimum values appear at a depth of ~ 2 km. At a depth of ~ 6.5 km feature C disappears and instead, two conductive blobs appear in the model. At a depth of ~ 3.4 km the conductive feature labeled D appears in the model which has a E-W extension. At shallower depths, this feature seems to be connected with feature A. However, at greater depths (> 6 km), both A and D separate and D takes the form of a blob, however, with a more pronounced E-W extension. The resistivity values of D have two minimums, at ~ 3.4 km and at ~ 20.6 km. The least pronounced structure in the model is the resistive feature labeled E. This feature appears at a depth of ~ 3.4 km to the east of A and south of D. The resistivity values of E increase gradually from $\sim 200 \Omega m$ to reach a maximum of $\sim 500 \Omega m$ at a depth of ~ 6.7 km.

To assess the ability of the model in Fig. 4.41 to fit the T_{zx} and T_{zy} data, the data fits at five selected periods are presented in Fig. 4.42 in terms of real parts of the measured and predicted induction vectors.

The data fit presented in Fig. 4.42 shows that at short and intermediate periods (1s and 45s) the data fit is acceptable within the used error bounds. Except at few stations which are affected by local noise. At a period of 181s, one can still observe an acceptable data fit, however only at the western and eastern stations. The induction vectors of stations located at the Al-Lisan peninsula show a considerable data misfit for all periods longer than 45s. At periods longer than 181s, the predicted induction vectors at the western stations start to point more southwards than the measured ones. At the longest period (2048s), an acceptable data fit can be observed at the western stations.

A comparison between the obtained conductivity distribution presented in Fig. 4.41 and the fault distribution in the study area reveals (Fig. 4.43):

- The N-S extending conductive structure (labeled A in Fig. 4.41) is located beneath the Dead Sea brines and delimited by the EBF at the eastern side. The western boundary of this conductive structure seems to coincide with the surface trace of the WBF. However, the lack of station coverage at the western side does not allow resolving the N-S extension of the WBF in the model. The thickness of this structure gives an indication that this structure does not only reflect the Dead Sea brines, since the maximum depth of the Dead Sea brines is only ~ 300 m. This structure might be correlated with the northern DSB which has a depth of ~ 5 km (Ben-Avraham & Shubert, 2006).
- The resistive structure (labeled B in Fig. 4.41) seems to correlate with the location and shape of the Al-Lisan salt diapir. However, due to the low sensitivity of the T_{zx} and T_{zy} data to detect vertical contrasts, the top of the Al-Lisan diapir seems to be deeper than inferred from geological and borehole information.

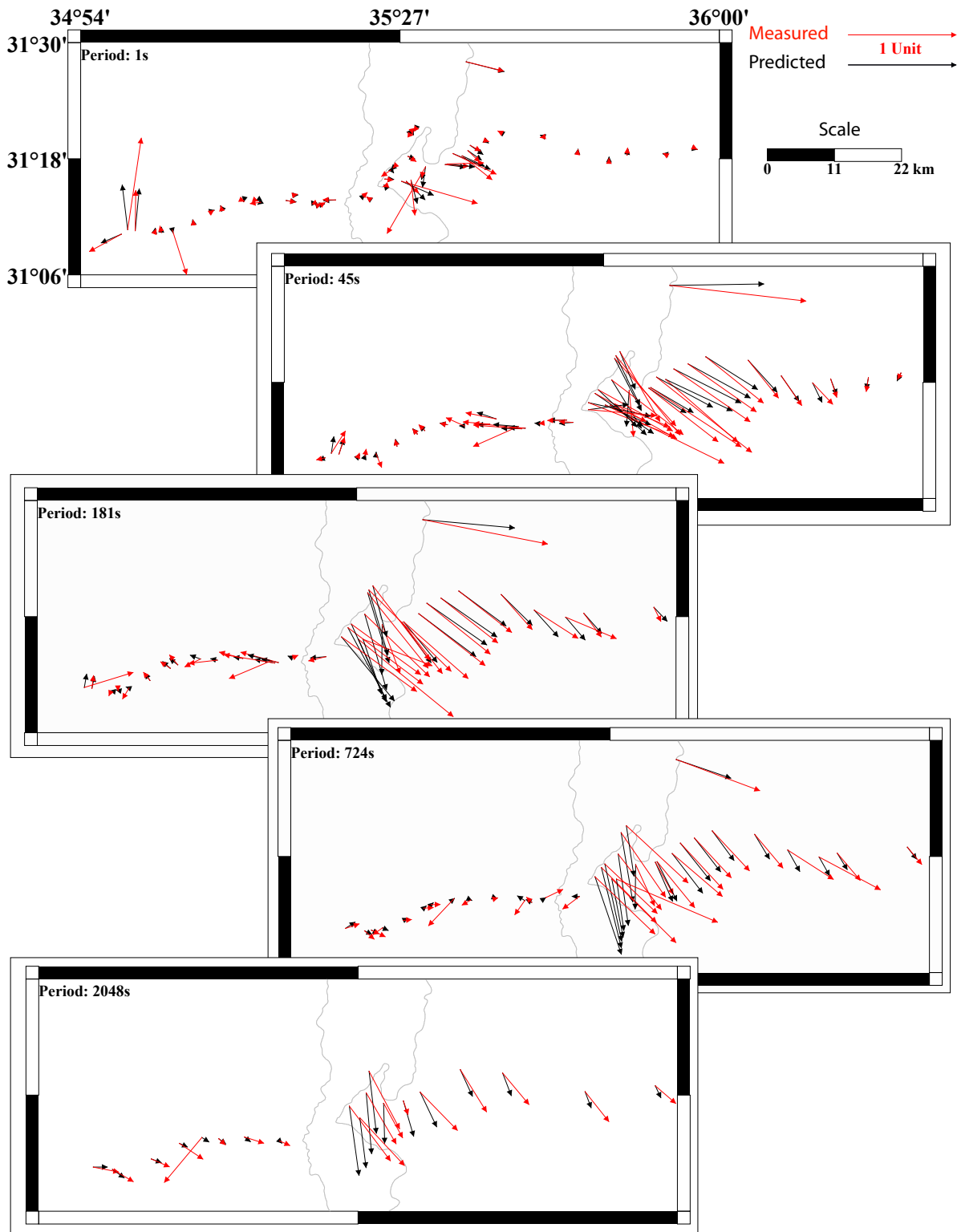


Figure 4.42: Comparison between real parts of the measured (red vectors) and predicted (black vectors) induction vectors using 3D inversion at five selected periods.

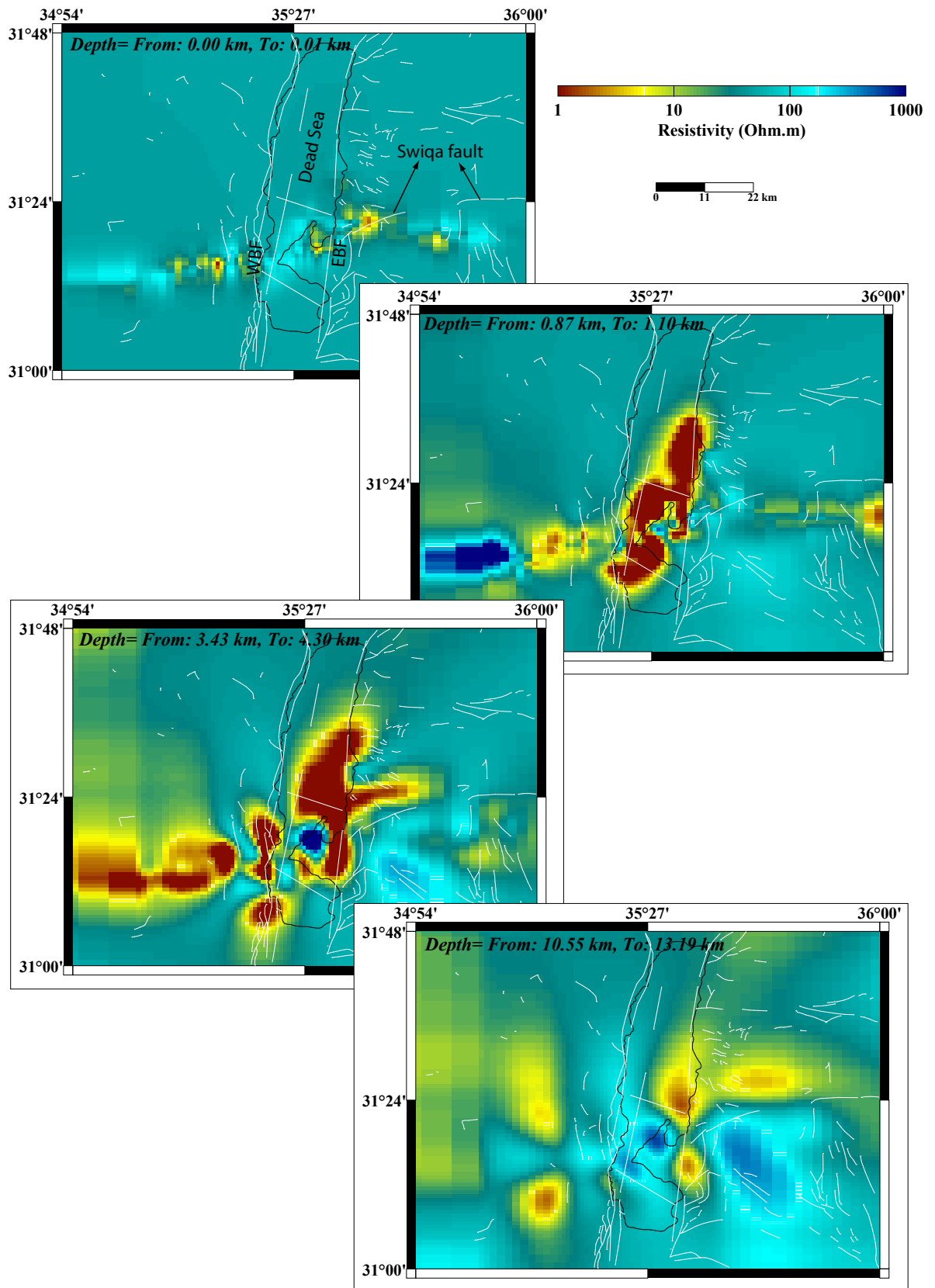


Figure 4.43: Comparison between the conductivity distribution obtained from fitting only the T_{zx} and T_{zy} data and the fault distribution in the study area.

- The horizontal contrast between the features labeled D and E in Fig. 4.41 seems to coincide with the trace of the Swiqa fault. An interpretation of the conductive structure D cannot be given, due to the lack of geological or other geophysical information from this area and at this depth. However, the location of the resistive feature E coincides with the location of the eastern resistive block indicated in the 2D model (Fig. 4.33), the Precambrian basement (Fig. 4.34) and the seismic basement (Fig. 4.36).

In the second 3D inversion run, the full impedance tensor is inverted. However, the main diagonal elements (Z_{xx} and Z_{yy}) are down weighted by adding high error floors because the values of these components are very small and noisy. The model used to invert the components Z_{xy} and Z_{yx} consists of $61 \times 72 \times 35$ model parameters in X, Y and Z directions, respectively, and 10 air layers. The model extends from ~ -700 km to ~ 700 km in both N-S and E-W directions and to a depth of ~ 950 km. A value of $100 \Omega m$ is assigned to all model parameters in the starting model. The same number of stations as previously is used. However, 27 periods and therefore 28 processors were used for the parallelized 3D inversion scheme.

The 3D model in Fig. 4.44 shows the most pronounced structures obtained from fitting the Z_{xy} and Z_{yx} components. The conductive zone labeled A in the model appears at the earth's surface and extends to a depth of ~ 8.5 km. However, at a depth of ~ 2 km the feature starts to split into northern and southern parts. Structure A has high conductivity values ($< 1 \Omega m$) in the upper part of the model (< 8 km). Whereas, the southern part is more conductive in the depth range between ~ 2 km and ~ 8.5 km. The resistive zone labeled B appears in the model at the same position as feature B in Fig. 4.41, however, it appears at a shallower depth (~ 200 m) and disappears at a depth of ~ 4 km. The resistivity values of element B increase from $\sim 200 \Omega m$ at ~ 200 m and reach values $> 1000 \Omega m$ at a depth of ~ 1 km. The conductive zone labeled C with resistivities of $\sim 1 \Omega m$ seems to extend only between ~ 1 km and ~ 2 km depths. The two resistive elements D and E appear in the model at different depths, however to the east and west of the conductive zone A. While D appears at a depth of ~ 1 km, E appears only at ~ 8.5 km. At greater depths (> 8.5 km), D and E appear in N-S direction and form two blocks at the eastern and western sides of the conductive feature A. In comparison to the result obtained from fitting only the T_{zx} and T_{zy} data (Fig. 4.41), the vertical layering is better resolved when inverting the Z_{xy} and Z_{yx} data (Fig. 4.44). For example, the tops of features A and B appear at shallower depths which is in good agreement with the results obtained from the 2D inversions and moreover with geological information which indicate e.g. the existence of the Al-Lisan salt diapir (the resistive feature B) at a depth of ~ 200 m. On the other hand, lateral conductivity contrasts like the transition from resistive zone D to conductive zone E in Fig. 4.41 are not resolved in the model of Fig. 4.44. To evaluate the fit of the Z_{xy} and Z_{yx} components of the model in Fig. 4.44, the

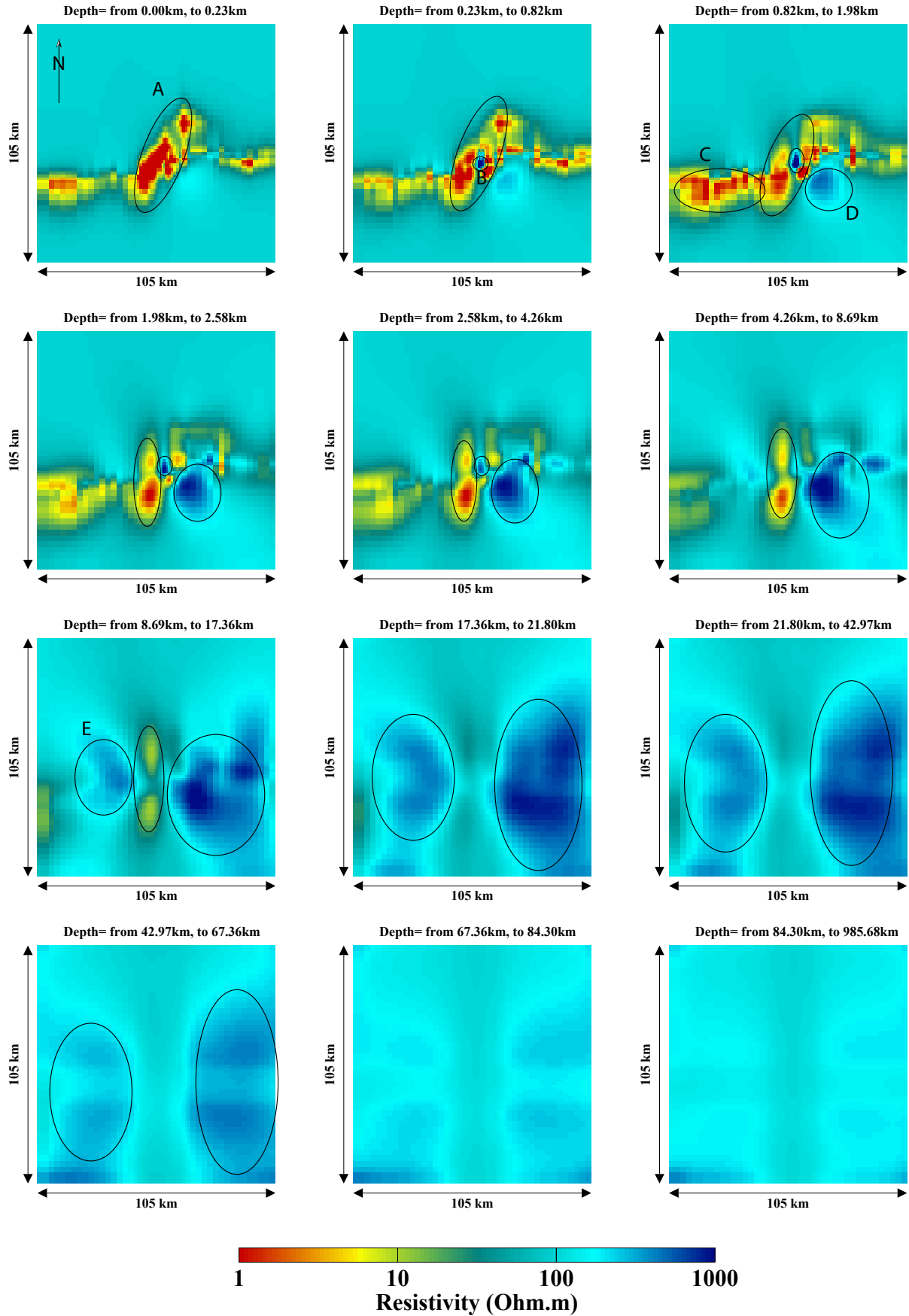


Figure 4.44: The 3D inversion result of fitting the components Z_{xy} and Z_{yx} , presented as depth slices. Features labeled A, B, C and D represent the most pronounced conductive or resistive zones in the model.

measured and predicted data of these components are presented in terms of apparent resistivities and phases at six stations distributed along the two profiles (Fig. 4.45).

A Comparison between the measured and predicted data shows that an acceptable data fit is reached, which means that the conductivity distribution of the model in Fig. 4.44 can explain the Z_{xy} and Z_{yx} data. Now, one of the reasons that guided me to invert the DESIRE-MT data in 3D is the fact that the 2D inversion could not explain the data at the central stations because they are influenced by 3D effects. How is the data fit in 3D in comparison to 2D at the central stations? Figure 4.46 shows a comparison between the data fit in 3D and 2D at one of the central stations (number 638). However, several points must be considered before comparing the data fit in 2D and 3D: i) Contrary to the 3D case, the data were rotated to the geo-electric strike direction for the 2D inversion. ii) For the 3D inversion, the components Z_{xy} and Z_{yx} were fitted, whereas the data components T_{zy} , ρ_{TM} , ϕ_{TM} , and ϕ_{TE} were used for the 2D inversion. iii) Different grid discretization and period layouts were used for the inversions in 2D and 3D. Nevertheless, a plausible comparison between the data fit in 2D and 3D can be achieved. Figure 4.46 shows clearly that the data fit is significantly improved in 3D. In particular when considering the phase curves in both 2D and 3D inversion.

For the inversion of DESIRE-MT data in 3D or 2D, the same data set were used which reflects the same underlying geology. This means that regional structures should appear in both 2D and 3D models. For this reason, it is worthwhile to compare the obtained 2D and 3D models in order to detect similarities and differences between the models (Fig. 4.47).

Figure 4.47 shows the preferred 2D model presented in Fig. 4.33) in comparison with two selected slices from the 3D model in Fig. 4.44 extending in E-W direction and the geological cross-section presented in Fig. 4.34. Since the stations of the main profile are not located on a straight line, the selected two slices from the 3D model are representative for either the western or the eastern stations (Fig. 4.47d). The comparison in Fig. 4.47 shows:

- The two resistive blocks in the east and west appear in the 2D and 3D models. However, the upper boundaries of the blocks are better resolved in 2D than in 3D. This is because different grid discretizations are used for the 2D and 3D inversion. In the 2D model the grid was finer discretized in Z direction. These two resistive blocks could be correlated with the crystallised igneous rocks of the Precambrian basement and could be considered as regional structures with a N-S extension. This is because, the 2D inversion, in which we assume that structures have infinite extension in N-S direction, was able to resolve these structures.
- The conductive structures at the western side which correspond to sedimentary formations of the Cretaceous and Jurassic appear in the 2D and 3D models.

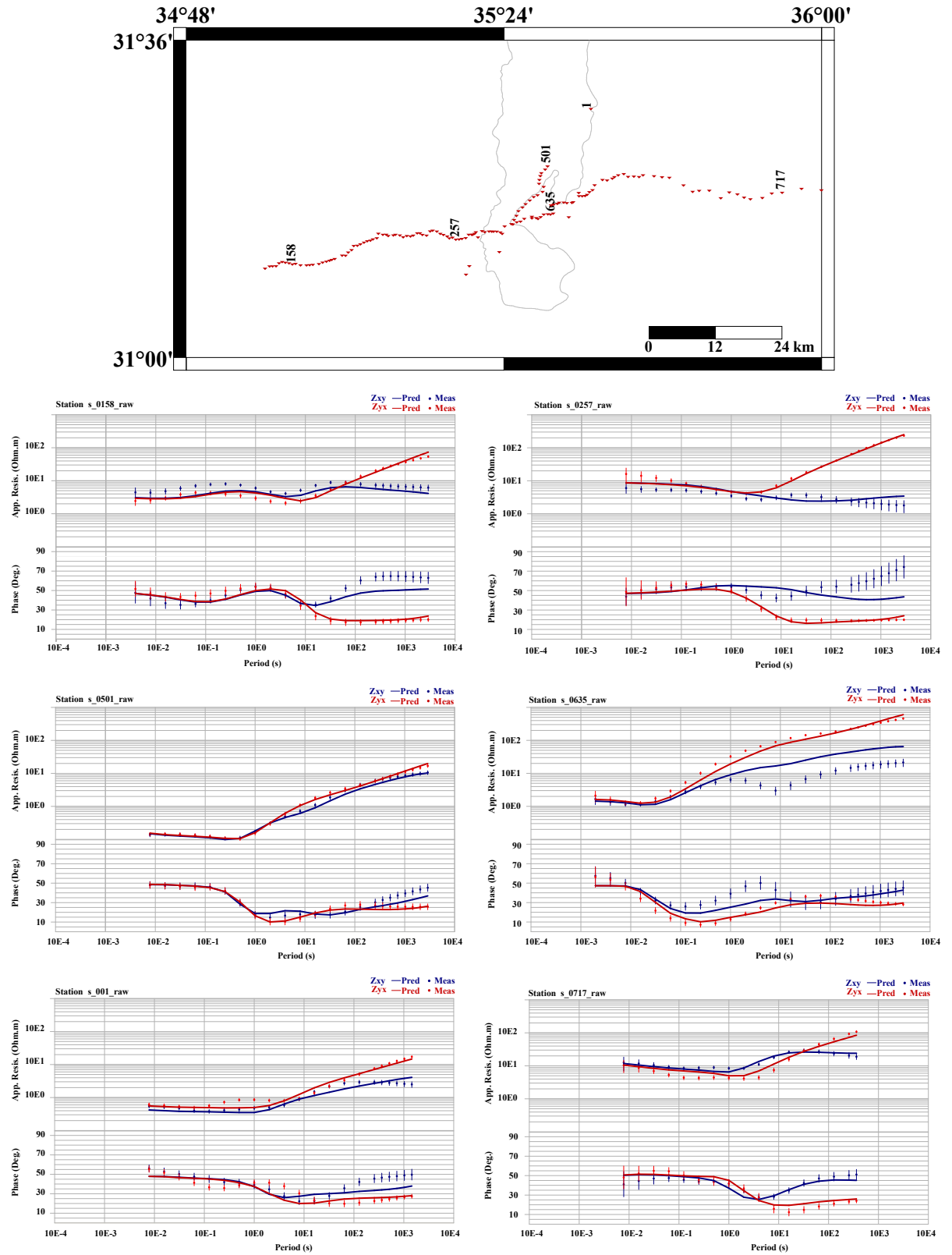


Figure 4.45: Comparison between measured (dots) and predicted (lines) data of the components Z_{xy} (blue) and Z_{yx} (red). The comparison reveals that an acceptable data fit is reached within the used error bounds.

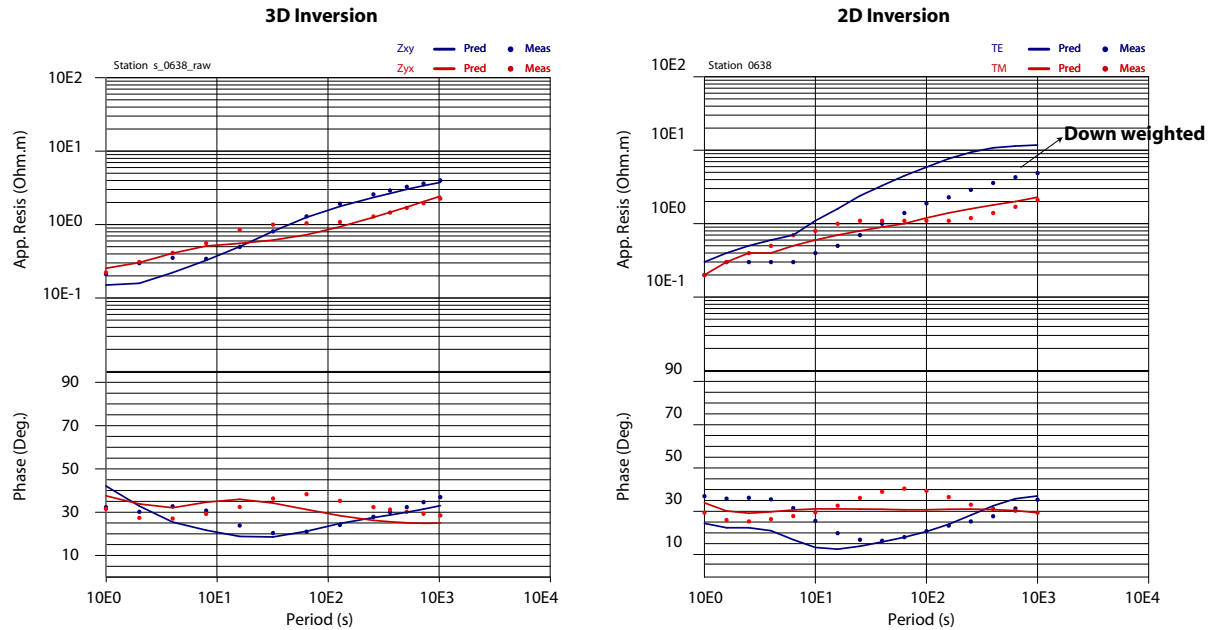


Figure 4.46: The data fit of station number 638 which is located in the central part of the study area obtained from 2D and 3D inversion. Significant improvements in the data fit obtained from the 3D inversion can be observed in particular in the phase curves. Please keep in mind that the measured data were rotated to the geo-electric strike direction (12°) for the 2D inversion.

However, in the 3D model (Fig. 4.47b) the structures seem to consist of only one conductive unit, whereas, in the 2D models they appear as two conductive layers. This could be caused by the coarser discretization of the 3D grid in Z direction. Because these structures appear in both models, they may be considered as regional features.

- At the eastern side, one can observe similarities between the structures at shallow depths. However, because of the poorer discretization at the eastern part of the 3D grid, it is difficult to make accurate comparison between the 2D and 3D models.
- The shallow resistive structure in the central part of the models which coincide with the location of the Halite intrusion (Al-Lisan diapir) is detected in both models. However, one can clearly see that the 3D inversion (Fig. 4.47b) has resolved this structure better than the 2D inversion. This is an obvious indication that this structure is a three dimensional feature with limited lateral extensions, particularly in N-S direction. Hence, this feature is more local structure which spans beneath the Al-Lisan peninsula. The stations from the in-basin profile (which extends in N-S direction) helped to resolve these structures in the 3D

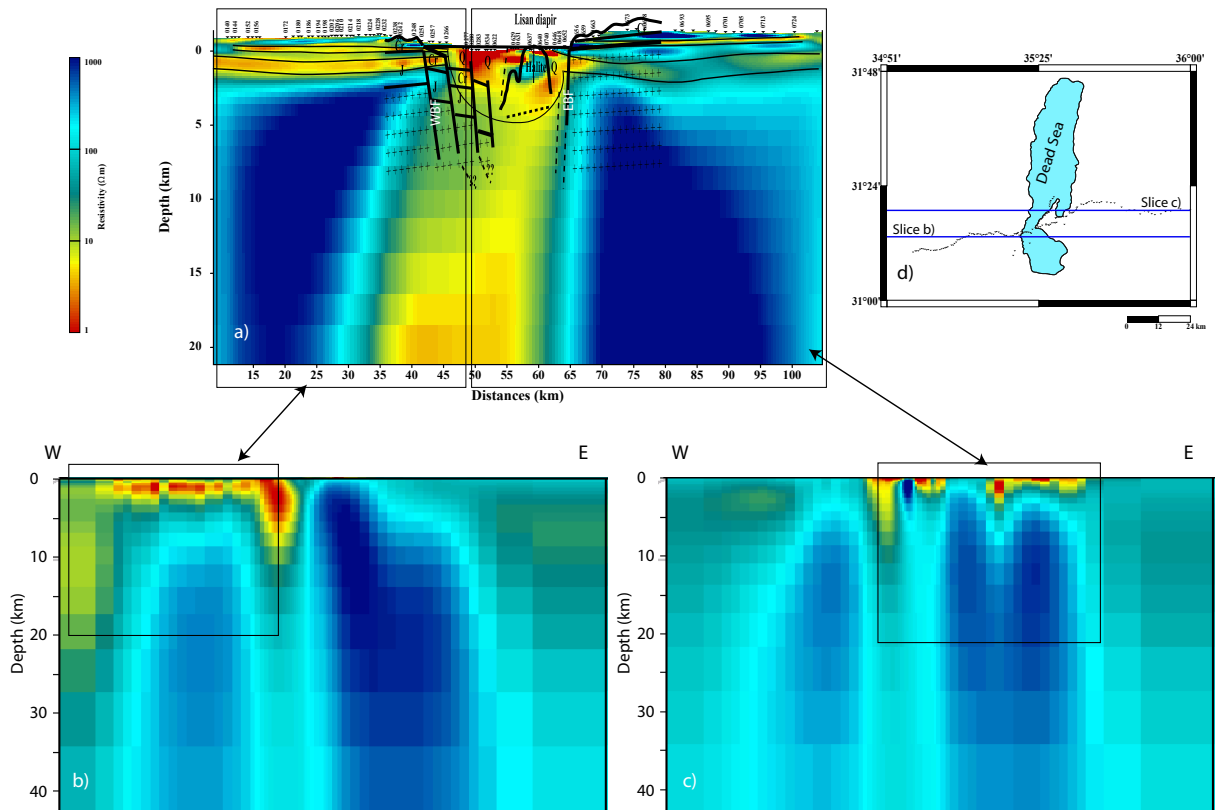


Figure 4.47: Comparison between a) the preferred 2D model, b) a slice from the 3D model extending in E-W direction following the western stations and c) a slice which is representative for the eastern stations. d) Map showing the location of the two slices as blue lines. The 2D model is presented together with the geological cross-section. The comparison reveals that structures like the Precambrian basement is reflected in both 2D and 3D models as two resistive blocks located to the east and west of the Dead Sea basin which can be considered as regional feature. Another structure that can be detected in the 2D and 3D models are the shallow conductive structures in the west, which in 3D appear as one layer, while in 2D as two layers. The Al-Lisan salt diapir which appears in both models as a resistive structure is better resolved in the 3D model, since it has a limited extension in N-S direction (local structure). The traces of the EBF and WBF can also be detected in both models. However, the trace of the WBF is less pronounced in the 2D model. The central shallow conductive structures can be also detected in both models, however, more pronounced in b). At the eastern side, conductive structure can be detected in the 2D and 3D models. The deep sub-vertical conductor that appears in the 2D model is less pronounced in the 3D model.

inversion.

- The central shallow conductive structures which are correlated with the Quaternary sedimentary formation can also be observed in the 2D and 3D models. However, these structures are more pronounced in the 3D model slice which is located to the south (Fig. 4.47b). These conductive structures extend deeper than in the 2D model. This could be an indication that the southern Dead Sea sub-basin is deeper than the northern sub-basin in support of the assumption of Ben-Avraham & Shubert (2006) (see Fig. 4.2).

Chapter 5

Conclusion and discussion

The work presented in my PhD thesis consists of two main parts:

Parallel computing

Parallel computations show their efficiency in solving huge numerical problems that occur in many natural sciences. For the magnetotelluric technique, the need to adopt parallel computing concepts is most obvious in modelling data in 3D. For the 3D modelling and inversion of MT data, a huge number of model parameters is needed to reasonably explain a complex, three dimensional subsurface, which means long run times and huge computer memory are required. However, by analysing and decomposing the mathematical formulations of the forward modelling and inversion problems in MT, I could show a novel and appropriate way to parallelize the computations.

The mathematical formulations for solving the forward modelling show that they have advantageous properties for parallelization. The main computation task when solving the forward modelling problem is the solution of a period dependent linear equation system that approximates Maxwell's equation on i.e. a staggered grid using i.e. the finite differences method. However, this equation system shows that:

1. The solution for different periods is independent which means that the equation system can be solved for each period separately.
2. For the 3D forward modelling in MT, two electric field polarisations (modes) are used to solve the equation system and to form the full impedance tensor (\mathbf{Z}). However, the solution for each polarisation can be computed separately.

Utilizing these two facts allowed me to parallelize the solution of the forward modelling problem in 3D over periods and modes by distributing the computation tasks between several processors. This means that each processor solves the equation sys-

tem for the electromagnetic (EM) field only for one period and one mode. In this case, not only a run time reduction can be achieved, but also less memory is required to save the EM field, as each processor needs memory to save the results corresponding to only one period. In the 3D forward modelling example used to demonstrate the efficiency of the parallel scheme, a speed-up factor of ~ 12 is reached.

In the data space Occam inversion, two computational tasks consume most of run time and computer memory when running a 3D inversion: The computation of the sensitivity matrix (\mathbf{J}) and the cross product $\mathbf{J}\mathbf{J}^T$. The sensitivity matrix has an advantageous structure that can be used for parallelizing its computation; it can be decomposed in blocks. Each block contains the sensitivity values corresponding to one period and one station. Since the computation of the values in each block does not depend on the values of the other blocks, I was able to parallelize the computation of \mathbf{J} over periods and stations. The computation of \mathbf{J} in a block-wise manner does not only reduce the time required but also the memory for saving \mathbf{J} , as each processor uses its own memory. However, this causes some difficulties in dealing with the next computational task.

Different strategies to calculate the cross product $\mathbf{J}\mathbf{J}^T$ for the distributed values of \mathbf{J} were discussed to emphasize the fact that parallelizing the computation of the cross product is not a straightforward procedure regarding the run time and communication. An optimal run time and reduced communication is reached by letting all participating processors work in a send/receive operation. Moreover, an optimal multiplication scheme which proved to be ideal for such multiplications is also discussed. For the non-linear conjugate gradient (NLCG) inversion scheme, an explicit use of \mathbf{J} is not required. However, multiplications of the type $\mathbf{J}^T\hat{\mathbf{d}}$ are required. In the presented parallel scheme, I considered this multiplication and parallelized its computation over periods. In total, I developed a parallel module which contains most of the time and computer memory consuming mathematical operations required in solving the 3D inverse problem of MT data.

The results of the generality (linking with different 3D inversion codes) and performance tests, can be summarised as follows:

- The developed parallelization and memory-efficiency concepts included in the parallel module have been linked to two different 3D MT inversion codes to convert them from serial to parallel codes. The conversion requires minimal changes in the parallel module regarding input and output.
- The run time required to solve the forward modelling problem depends mainly on the structures included in the model and the used periods. The overall parallel run time is determined by the slowest processor which is working with the period that requires the maximum number of quasi minimum residual (QMR) iterations.

- The overall speed-up factor which could be achieved in solving the forward modelling problem for a certain test model and period layouts has a value of ~ 12 .
- For the 3D inversion test, the data space Occam inversion requires only 6 iterations to converge to the desired root mean square (RMS), whereas 50 iterations were required by the NLCG. Nevertheless, the overall run time for the inversion was shorter using NLCG than data space Occam inversion. This is because the total run time for the NLCG is dominated by the operation $\mathbf{J}^T \hat{\mathbf{d}}$ which consumes less run time in comparison to computing the sensitivity matrix and achieving the multiplication $\mathbf{J}\mathbf{C}_m\mathbf{J}^T$ for the data space Occam inversion.

DESIRE-MT

In collaboration with colleagues from Germany, Jordan and Israel, magnetotelluric measurements along two profiles were carried out to investigate the deep structures of the southern Dead Sea basin (DSB). The MT data were collected at 150 sites distributed along an east-west (main) profile and a north-south (in-basin) shorter profile. I used a combination of single site and remote reference data processing to estimate transfer functions with high to acceptable data quality.

The subsequent geo-electric strike direction and dimensionality analysis of the data set reveals that the study area along the profile can be subdivided into three segments: The eastern and western segments have a close to 2D nature with a stable geo-electric strike direction of $\sim \text{N}10^\circ\text{E}$ which is in agreement with the main axis of the Dead Sea basin. The strike directions obtained for the central segment within the basin are scattered. Therefore, I cannot infer a preferred direction which indicates that the basin has predominately 3D characteristics. Nevertheless, the induction vectors suggest the existence of a N-S extending conductive zone in the central part of the study area. The behaviour of the induction vectors at long periods, however, indicates that this structure could extend several kilometres in depth.

Using an available 2D code (Rodi & Mackie, 2001), I computed a series of 2D inversion models from the DESIRE-MT data along the main profile. Fitting the MT data in 2D for separate modes emphasises their sensitivity to different structures. Finally, a joint 2D inversion shows that there exists a number of stable and reliable structures that are required by the data and which are independent from inversion parameters and setting. However, the 2D inversion produced also some structures which seemed to be artifact. With resolution studies using constraint inversion I was able to identify inversion artifact. The overall data fit for the preferred 2D conductivity model is overall acceptable, however, the central stations are poorly fitted in all inverted models which could already be suspected based on the results from the geo-electric strike direction and dimensionality analyses.

The apparently three-dimensional and more complex structure in the basin in addition to the observed induction vectors, reveal a necessity for a 3D interpretation (modelling and inversion) of the DESIRE-MT data. For the 3D inversion, I used the developed parallel scheme which was linked to the modular system of G. Egbert. The DESIRE-MT data were inverted individually using the vertical magnetic components and both off-diagonal components of the impedance tensor (\mathbf{Z}). Two main points must be considered when discussing the results from the 3D inversion: i) The station set-up used for the DESIRE-MT was not ideal for 3D inversion, because the majority of stations are distributed along the main profile. Only in the central part (within the basin), where the main and in-basin profiles cross we do have good spatial station coverage. ii) The modular system code used for the 3D inversion is still being developed and tested. So far, the application of modular 3D inversion was only to synthetic data. In my work, its the first time that this code is applied to real MT data. Moreover, many parameters which control the inversion run have not been fully tested to date.

To compare the conductivity distribution obtained from 2D and 3D inversion and discuss it in a geological context, I compared the final 2D and 3D inversion models with geological cross-sections and stratigraphical columns available from the Dead Sea basin and its direct vicinity. In the following I will discuss the resolutions of the different inversions and compare the prominent conductivity features with tectonic and geological information:

- As an introductory comment with regard to the 3D inversion, I would like to point out that I used the vertical magnetic transfer function and both off diagonal components of the impedance tensor, as the diagonal components were only resolved for longer periods. In restricting to off-diagonal elements the coordinate systems becomes of importance again, similarly to a 2D inversion. For 3D inversion I used the MT data in an unrotated coordinate system, which means a deviation from the geo-electric strike direction by 10° . This does not have a strong influence, however, it is desirable to invert the full tensor in the future.
- The Al-Lisan salt diapir known from geological cross-section and the Al-Lisan1 deep borehole coincides spatially with a shallow resistive structure in the central parts of the 2D and 3D models. The relatively high resistivity values of this structures indicates that the Halite intrusion has a low porosity and that the overlying Dead Sea brines did not penetrate. This feature, however, is better resolved by 3D inversion than by the 2D model because the Al-Lisan diapir is a local 3D feature. In the 2D model, the upper boundary of the Al-Lisan diapir is resolved, but not its spatial extension. In particular the depth extension which seems to be terminated by conductive structures at a depth of ~ 2 km, disagrees with geological and borehole information. It is possible that the influence of

the northern and southern Dead Sea brines is expressed in the 2D model as a conductive structure at the central shallow part. In the obtained 3D model, however, the depth to the top of the Al-Lisan diapir and its vertical extension to ~ 4.5 km are better resolved, in particular when fitting impedance tensor \mathbf{Z} rather than the vertical magnetic field. This implies that the influence of the northern and southern Dead Sea brines which can be modelled in 3D, is strong. Moreover, the shape of the Al-Lisan diapir in the 3D conductivity model is much more similar to the shape inferred by the velocity model presented by [Al-Zoubi & Brink \(2001\)](#).

- The stratigraphical column of the Zohar-8 deep borehole provides detailed information about the layering at the western flank of the Dead Sea basin, which is well resolved in the 2D conductivity model. In particular, I would like to point out a coincidence of the lower Cretaceous Kurnub formation with a conductive layer. Hydrologists and geologists in Israel define the Kurnub formation as a regional aquifer, which would be in agreement with the observed electrical conductivity values. The underlying upper Jurassic formations (Zohar and Sherif) coincide with a resistive layer. These two formations are supposed to contain organic matter with high concentration of natural gas. This could explain the relatively high resistivity found for this layer that sandwiched between two conductive layers. The transitions from Jurassic to Triassic as well as from Triassic to Permian are not well resolved. Instead, a conductive layer extends from middle Jurassic to the Precambrian. Further in depth, the transition from the sedimentary formations presented in the Cretaceous, Jurassic, Triassic and Permian to the crystalline igneous rocks of the Precambrian is resolved in the model as a transition from the upper alternating resistive-conductive sequences to the basement. The coarse discretization of the 3D model makes the layered structures merge into only one conductive layer. For the same reason, the upper boundaries of the eastern and western crystalline Precambrian basement rocks are better resolved in the 2D model than in the 3D models.
- Only little geological information is available from the Jordanian side, making the interpretation of the conductivity model more difficult. However, some hydrogeological information could be used to correlate the uppermost conductive layer in the 2D model with the Balqa formation which is defined by hydrologists and geologists in Jordan as a regional aquifer. Again, due to the coarse discretization of the 3D model, this feature is better resolved in the 2D model.
- The surface trace of the EBF is well resolved in the 2D and 3D models in terms of a lateral contrast between a resistive eastern block and a conductive central structure. In the 3D model, however, the central conductive structure is less pronounced. Instead, shallow conductive structures extending in N-S direction

which coincide with the location of the Dead Sea brines and the underlying sedimentary basin fill can be observed in the 3D model. The 3D inversion model as well as 3D forward modelling studies conducted to analyse the behaviour of the induction vectors reveal that the thickness of the N-S extending central structure is ~ 17 km. In particular the 3D model based on fitting the off-diagonal elements of \mathbf{Z} indicates that the southern part is deeper and more pronounced in terms of high conductivity values. This finding would be in agreement with the assumptions of Ben-Avraham & Shubert (2006) that the southern sub-basin is deeper than the northern sub-basin.

- Only with the assumption that the WBF has a similar behaviour as the EBF, the surface trace of the WBF is better resolved in the 3D models. This is because, the same lateral contrasts can be observed at the EBF and WBF in the 3D model which are not similar in the 2D model. In the 2D model, one can observe a deep sub-vertical conductive structure which seems to be smeared out with overlying shallow conductive structures in the central part and delimited between the eastern and western resistive blocks. Resolution studies on this feature in 2D models indicates a depth extension of this structures at least to the mid-crust.
- The sedimentary formation of the Quaternary in the basin correlates with highly conductive structures extending at least to a depth of ~ 3 km in the 2D model. This anomaly could be caused by the Dead Sea brines penetrating several kilometres deep into the sedimentary formations due to low porosity. Since this feature smears out with the underlying conductive deep sub-vertical structures a distinct separation between the two conductive structures and therefore determining the thickness of the basin is difficult in the 2D model.

As pointed out above, the 3D model based on fitting the off-diagonal elements of \mathbf{Z} reveals the existence of two conductive structures that correlate with the northern and southern sub-basins. However, in the central part beneath the Al-Lisan peninsula, there is no indication of a deep reaching sub-vertical conductor in the 3D model. One explanation for the existence of a deep reaching sub-vertical conductor in the 2D model might be the influence of the northern and southern sub-basins on the data, which is then expressed as a deep sub-vertical conductor, since modelling of N-S terminated structures is not possible in the 2D inversion. At this point we have to discriminate between the basin filling and the border faults. Because of the high conductivity beneath the basin we cannot tell two scenarios apart: (i) the border faults show up as sub-vertical zones of high conductivity, merged with the conductive filling of the basin or (ii) the border faults are not associated with a sub-vertical zone of high conductivity and only the basin exhibits high electrical conductivities. In the first case, the images of the border faults could be compared with other large fault systems.

In literature we often find a comparison of the Dead Sea Transform (DST) with the San Andreas Fault (SAF) in California, in particular if we extend our focus to the entire DST.

At the SAF geophysical, geological and geochemical studies across this major strike-slip fault system showed that the dynamics and earthquake generation processes are closely linked with (over-pressurized) fluids (Zoback *et al.* , 1987; Becken *et al.* , 2008). In addition to meteoric water and formation water, which possibly cause a shallow fault zone conductor, there is evidence for fluids originating from the earth's mantle, forming a conductive channel from the lower crust or even the upper mantle to shallow levels. Ritter *et al.* (2005) compared the shallow electrical signatures characteristic of four large fault systems (including the DST at a different location) and suggested that most large fault systems are characterized by zones of high conductivity extending to a depth of several kilometres (< 10 km). The high conductivity of active faults in the upper part is attributed to fluids within a highly fractured damage zone. However, Ritter *et al.* (2003 2005) found that the Araba segment of the DST, which they investigated, shows a lack of a fault zone conductor in the upper crust and seems to act as an impermeable barrier to cross-fault fluid transport. These authors suggested that strain may have been localized for a considerable time span along a narrow, metre-scale damage zone with a sustained strength difference between the shear plane and the surrounding host rock. Obviously, the situation at the extensional pull-apart Dead Sea basin is markedly different, as the DST separates into two sub-parallel border faults. Similarly to the behaviour of the DST farther south, we do not observe a shallow fault zone conductor located beneath the trace of both border faults. However, we cannot completely exclude such conductors as they may be shielded by the very conductive basin filling. On the other hand, the image of deep reaching conductivity anomaly beneath the basin, somehow resembles the SAF near Parkfield, where a deep reaching sub-vertical zone of high conductivity may provide a pathway for fluids from the earth's mantle.

The results obtained from the 3D forward modelling and inversion of DESIRE-MT data do not only show a good correlation between the 2D and 3D models, but also that there are additional structures that cannot be resolved using a 2D approach. In particular, I want to mention an off-profile conductive structure extending in E-W direction located north of the western part of the main profile on the Jordanian side which appears in the 3D model in a depth range of ~ 3.4 km to more than 20 km. The southern boundary of this conductive feature is located near the surface trace of the Swiqa fault. However, due to the lack of MT sites in this area as well as geological and other geophysical information, an interpretation of these deep conductivity structures cannot be given.

Typically maps of induction vectors nicely picture of off-profile features. Using regional 3D models including the Mediterranean Sea and the red Sea reveals that the behaviour of the induction vectors at longer periods is strongly affected by the NE-SW extending Mediterranean Sea.

In addition to the comparison with geological information, I also compared the electrical conductivity 2D model with the seismic 2D P-wave velocity model obtained from the DESIRE wide-angle seismic profile. It reveals a coincidence of the top of the seismic basement with the top of both, the eastern and western resistive blocks. However, the top of the seismic basement in the basin defined in the 2D P-wave as a horizontal discontinuity could not be inferred from the conductivity model. Because of the reduced sensitivity of the wide-angle seismics to shallow small-scale structures, the uppermost 5 km cannot be compared.

In summary, the work presented in my thesis shows the advantages and disadvantages of 2D and 3D inversion. I am convinced that 3D inversion is the way forward, but (i) we need data sets with a good areal coverage, (ii) we have to thoroughly compare 2D and 3D inversion results, (iii) we have to understand and evaluate different inversion strategies, similar to what has been done for 2D inversions and (iv) further improving 3D inversion algorithms.

Zusammenfassung und Diskussion

Die vorgelegte Arbeit in meiner Dissertation besteht aus zwei Hauptteilen:

Parallelisierte Löser

Parallelisierte Löser sind besonders effektiv bei der numerischen Lösung von großen Gleichungssystemen mit vielen Unbekannten, wie sie in den Naturwissenschaften häufig bei der diskreten Behandlung von Differentialgleichungen auftreten. In der Magnetotellurik ist der Bedarf an Parallelisierungskonzepten bei der 3D Modellierung besonders offensichtlich. Zur 3D Modellierung und Inversion von MT-Daten wird eine große Anzahl von Modellparametern benötigt, um einen komplexen, dreidimensionalen Untergrund hinreichend genau darstellen zu können. Dies führt zu langen Rechenzeiten und hohem Speicherbedarf. Durch Analyse und Zerlegung der mathematischen Formulierung des Vorwärts- und Inversionsproblems der MT konnte ich jedoch einen neuen, effizienten Weg zur Parallelisierung der Berechnungen aufzeigen und implementieren. Die mathematischen Formulierungen zur Lösung der Vorwärtsmodellierung haben für die Parallelisierung vorteilhafte Eigenschaften. Die Hauptrechenaufgabe bei der Lösung des Vorwärtsproblems ist die Lösung eines periodenabhängigen, linearen Gleichungssystems, das die Maxwellgleichungen zum Beispiel auf einem Blockgitter nach der Finite-Differenzen-Methode annähert. Dieses Gleichungssystem hat folgende Eigenschaften:

- Die Lösung für verschiedene Perioden ist unabhängig voneinander, d. h. das Gleichungssystem kann für jede Periode gesondert gelöst werden.
- Bei der 3D Vorwärtsmodellierung in der MT werden zwei unabhängige Polarisationen des elektrischen Feldes verwendet, um das Gleichungssystem zu lösen und den vollständigen Impedanztensor (\mathbf{Z}) zu bilden. Die Lösungen für die einzelnen Moden können voneinander getrennt berechnet werden.

Die Verwendung dieser beiden Eigenschaften erlaubte mir, die Lösung der Vorwärtsmodellierung in 3D über die einzelnen Perioden und Polarisationen zu parallelisieren

und die Rechenaufgaben auf mehrere Prozessoren zu verteilen. Jeder Prozessor löst das Gleichungssystem für das elektromagnetische (EM) Feld für nur eine Periode und Polarisationen. Dadurch wird die effektive Rechenzeit erheblich reduziert; ausserdem ist relativ wenig Arbeitsspeicher pro Knoten erforderlich da jeder Prozessor nur die EM-Felder für eine Periode speichern muss. In einem Geschwindigkeitsvergleich zwischen der seriellen und der massiv parallelisierten 3D-Vorwärtsmodellierung wurde eine Verkürzung der Rechenzeit um den Faktor 12 ermittelt. Bei der sogenannte "Data space" 3D-Occam-Inversion liegt der größte Rechenaufwand und Speicherbedarf bei zwei Aufgaben: die Berechnung der Sensitivitätsmatrix (\mathbf{J}) und ihres Kreuzproduktes $\mathbf{J}\mathbf{J}^T$. Die Sensitivitätsmatrix verfügt über eine vorteilhafte Struktur, auf die zur Parallelisierung der Berechnungen zurückgegriffen werden kann: Sie kann in Blöcke zerlegt werden. Jeder Block enthält die zu einer Periode und Station gehörenden Sensitivitäten. Da die Berechnung der Werte in einem Block nicht von den Werten anderer Blöcke abhängt, war es mir möglich, die Berechnung von \mathbf{J} über Perioden und Stationen zu parallelisieren. Die blockweise Berechnung der Sensitivitätsmatrix reduziert nicht nur die Berechnungszeit sondern auerdem den Speicherbedarf pro Knoten, da jeder Prozessor seinen eigenen Speicher verwendet. In der Arbeit werden verschiedene Strategien zur Berechnung des Kreuzproduktes $\mathbf{J}\mathbf{J}^T$ für die verteilten Blöcke von \mathbf{J} wurden diskutiert, da die in Hinblick auf Rechenzeit und Kommunikation zwischen den Nodes beste Strategie zur parallelisierten Berechnung des Kreuzproduktes nicht offensichtlich ist. Hier ist eine Abwägung zwischen Kommunikation und eigentlicher Rechenzeit zu treffen. Die optimale Rechenzeit mit minimaler Kommunikation wurde erzielt, indem alle beteiligten Prozessoren in einem Sende-/Empfang-Modus arbeiten, d.h. alle Prozessoren tauschen untereinander die zur Berechnung notwendigen Daten aus. Darüber hinaus wurde ein optimales Multiplikationsschema zur Berechnung des Kreuzproduktes diskutiert. Bei Anwendung des nicht-linearen konjugierten Gradienten (NLCG) Inversionschemas ist eine explizite Evaluierung der Einträge in \mathbf{J} nicht erforderlich. Jedoch treten Matrix-Vektor-Multiplikationen des Typs $\mathbf{J}^T \hat{\mathbf{d}}$ auf. In dem dargestellten Parallelisierungsschema habe ich diese Multiplikationen über die Perioden parallelisiert. Zusammenfassend habe ich somit ein Parallelisierungsmodul entwickelt, das die meisten zeit- und speicheraufwändigen mathematischen Operationen parallelisiert und beschleunigt. Die Ergebnisse der Allgemeinheit des Parallelisierungsmoduls (ausprobiert mit zwei verschiedenen 3D inversion Codes) und Leistungstests können folgendermaßen zusammengefasst werden:

- Das entwickelte Parallelisierungs- und Speichereffizienz-Konzept des Parallelisierungsmoduls wurde mit zwei verschiedenen, seriellen 3D MT-Inversionsprogrammen verknüpft, um parallele Berechnungen zu ermöglichen. Diese Umwandlung erfordert lediglich minimale Abänderungen der Parallelmodule in Bezug auf Ein- und Ausgabeformate.
- Die Rechenzeit zur Lösung des Vorwärtsproblems hängt im Wesentlichen von der

Komplexität des Modells und der Anzahl der verwendeten Perioden ab. Die Gesamtlaufzeit der parallelisierten Lösung wird von der Berechnungszeit für die Periode mit der größten Anzahl an Quasi-Minimum-Residuum Iterationen bestimmt.

- Die maximal erzielte Beschleunigung bei der Lösung des Vorwärtsproblems für ein gegebenes Modell und Periodenlayout liegt bei Faktor 12.
- Bei den 3D Inversionstests benötigt die Occam-Inversion lediglich 6 Iterationen, um den gewünschten RMS zu erreichen, wohingegen der NLCG-Algorithmus 50 Iterationen erfordert. Dennoch liegt die benötigte Rechenzeit des NLCG-Algorithmus unter der der Occam-Inversion. Dies ist darauf zurückzuführen, dass die Laufzeit des NLCG von der Operation $\mathbf{J}^T \hat{\mathbf{d}}$ bestimmt wird, die offenkundig deutlich weniger Zeit in Anspruch nimmt als die Multiplikation $\mathbf{J}\mathbf{C}_m\mathbf{J}^T$, welche für die Occam-Inversion benötigt wird.

DESIRE-MT

In Zusammenarbeit mit Kollegen aus Deutschland, Jordanien und Israel wurden entlang von zwei Profilen magnetotellurische Messungen durchgeführt, um die Tiefenstruktur des südlichen Tote-Meer-Beckens (DSB) zu untersuchen. Die MT-Daten wurden an 150 Stationen entlang eines in Ost-West-Richtung verlaufenden Hauptprofils und eines kurzen Querprofils innerhalb des DSBs gemessen. Bei der Auswertung der Daten verwendete ich eine Kombination von Single-Site- und Remote-Reference-Processing, um Übertragungsfunktionen von annehmbarer Qualität zu ermitteln. Die Ergebnisse der anschließenden geoelektrischen Streichrichtungs- und Dimensionalitätsanalysen ermöglichen eine Unterteilung des Hauptprofils in drei Abschnitte: Die Daten östlichen und westlichen Bereich weisen in guter Näherung ein 2D-Verhalten auf, mit einer stabilen geoelektrischen Streichrichtung von etwa N10E, die mit der Lage der Hauptachse des Tote-Meer-Beckens übereinstimmt. Im mittleren Abschnitt, der im Wesentlichen den Bereich innerhalb des Tote-Meer-Beckens umfasst, streuen die Werte für die Streichrichtung hingegen sehr stark, so dass sich für diesen Bereich keine eindeutige Streichrichtung ableiten lässt. Dies zeigt an, dass der Untergrund in diesem Profilschnitt stark von 3D-Strukturen geprägt ist. Die Induktionspfeile deuten die Existenz einer gut leitfähigen Zone an, die sich in Nord-Süd-Richtung durch den zentralen Bereich des Messgebietes erstreckt. Das Verhalten der Induktionspfeile bei langen Perioden lässt eine vertikale Ausdehnung dieser Struktur bis in mehrere Kilometer Tiefe vermuten. Unter Verwendung des 2D-Codes von [Rodi & Mackie \(2001\)](#) berechnete ich eine Reihe von 2D Inversionsmodellen entlang des Hauptprofils. Getrennte Inversionen der einzelnen MT-Moden veranschaulichen die Sensitivität der Moden für jeweils unterschiedliche Strukturen. Eine gemeinsa-

me Inversion aller Moden zeigt, dass es eine Reihe von stabilen und verlässlichen Strukturen gibt, die von den Daten gefordert werden und unabhängig von Inversionsparametern sind. Einige die von der 2D Inversion eingeführten Strukturen konnte ich mit Hilfe von Auflösungsstudien und Zwangsbedingungen in der Inversion als Artefakte identifizieren. Die Datenanpassung für das bevorzugte 2D-Leitfähigkeitsmodell ist insgesamt akzeptabel, jedoch werden die Stationen im mittleren Profilabschnitt von allen 2D Inversionsergebnissen nur unzureichend angepasst. Dies war bereits auf Grund der Ergebnisse Streichwinkelanalyse zu vermuten. Die offensichtlich komplexe, dreidimensionale Struktur des Beckens und die beobachteten Induktionspfeile machen eine 3D Interpretation (Modellierung und Inversion) der DESIRE-MT-Daten erforderlich. Für die 3D Inversion verwendete ich das entwickelte Parallelisierungsschema in Verbindung mit dem Modularen System von G. Egbert. Zunächst invertierte ich die Daten komponentenweise getrennt: die magnetischen Komponenten sowie beide Nebendiagonalelemente des Impedanztensors (Z). Folgende zwei Punkte müssen bei der Diskussion der 3D-Inversionsergebnisse berücksichtigt werden: i) Die Stationsverteilung des DESIRE-MT-Projektes war nicht ideal für 3D-Inversionen, da der Großteil der Stationen entlang eines Hauptprofils verteilt ist. Nur im mittleren Bereich innerhalb des Beckens, wo Haupt- und Querprofil sich kreuzen, liegt eine gute räumliche Stationsüberdeckung vor. ii) Der Code des Modularen Systems befindet sich noch in der Entwicklungs- und Testphase; die Anwendung der modularen 3D-Inversion beschränkte sich daher bisher auf synthetische Daten. Im Rahmen meiner Arbeit wurde diese Software das erste Mal auf im Feld gemessene Daten angewendet. Die verschiedenen Parameter, die die Inversion kontrollieren, und ihre Einflüsse sind daher zurzeit noch nicht in vollem Umfang getestet worden. Ich habe die endgültigen 2D und 3D Inversionsergebnisse mit geologischen Profilen und stratigraphischen Ergebnissen aus dem Tote-Meer-Becken und der nahen Umgebung verglichen. Im Folgenden diskutiere ich das Auflösungsvermögen der verschiedenen Inversionsergebnisse und vergleiche markante Leitfähigkeitsstrukturen mit tektonischen und geologischen Informationen:

- Zunächst ist zu den 3D-Inversionen anzumerken, dass ich als anzupassenden Größen die vertikalen magnetischen Übertragungsfunktionen und die Nebendiagonalelemente des Impedanztensors Z gewählt habe. Die Diagonalelemente des Impedanztensors waren nur für die längeren Perioden hinreichend gut bestimmbar. Durch die Beschränkung auf die Nebendiagonalelemente von Z ist - ähnlich wie bei 2D-Inversionen - die Wahl des Koordinatensystems von Bedeutung. Zur 3D-Inversion verwendete ich unrotierte MT-Daten, was einer Abweichung des Koordinatensystems gegenüber der geoelektrischen Streichrichtung von 10° entspricht. Dies hat keinen großen Einfluss auf die Ergebnisse. Es ist jedoch wünschenswert, zukünftig den vollen Impedanztensor zu invertieren.
- Die aus dem geologischen Profil und der Al-Lisan-Bohrung bekannte Lage des Al-Lisan-Salzdiapirs fällt mit einer oberflächennahen, schlecht leitfähigen Struk-

tur im Zentrum der 2D- und 3D-Modelle zusammen. Die relativ hohen Widerstandswerte dieser Struktur deuten an, dass die Halitintrusion kristallisiert ist und über eine niedrige Porosität verfügt, so dass die darüber befindliche Sole des Toten Meers nicht in den Diapir eindringen kann. Diese Struktur ist in den 3D-Modellen besser aufgelöst als in den 2D-Modellen, da der Al-Lisan-Diapir dreidimensional ist. Im 2D Modell ist die obere Begrenzung des Al-Lisan-Diapirs aufgelöst, nicht jedoch seine räumliche Ausdehnung. Insbesondere die suggerierte vertikale Ausdehnung, die durch gut leitfähige Strukturen in etwa 2 km Tiefe beschränkt scheint, steht im Widerspruch zu geologischen Informationen und Bohrlochdaten. Dies können damit zusammenhängen, dass der Einfluss der nördlichen und südlichen Solen des Toten Meers im 2D-Modell als leitfähige Struktur im mittleren, oberflächennahen Bereich abgebildet wird. Im 3D-Modell werden die obere Begrenzung des Al-Lisan-Diapirs und seine vertikale Ausdehnung von etwa 4.5 km besser aufgelöst, insbesondere bei Anpassung der Impedanzensorelemente. Dies zeigt den großen Einfluss der nördlichen und südlichen Solen des Toten Meers, die in 3D modelliert werden können. Darüber hinaus ist die Gestalt des Al-Lisan-Diapirs im 3D-Leitfähigkeitsmodell der Form des Diapirs, die aus dem seismischen Geschwindigkeitsmodell von [Al-Zoubi & Brink \(2001\)](#) abgeleitet wurde, sehr viel ähnlicher.

- Am Westrand des Toten Meeres, welcher durch das 2D-Leitfähigkeitsmodell besonders gut aufgelöst werden konnte, liegen detaillierte stratigraphische Information aus der Zohar-8-Tiefbohrung vor. Hier möchte ich insbesondere auf die Kurnub-Formation (untere Kreide) eingehen, welche elektrisch als guter Leiter in Erscheinung tritt. Israelische Hydrologen und Geologen beschreiben die Kurnub-Formation als regionalen Aquifer. Diese Interpretation wird durch unser Leitfähigkeitsmodell unterstützt. Die oberjurassische Zohar- und Sharif-Formation im Liegenden der Kunrub-Formation fallen mit einem schlechten Leiter zusammen. Diese beiden Formationen enthalten vermutliche organische Anteile, evntl. mit einen relativ hohen Anteil an Erdgas, was die geringen elektrischen Leitfähigkeiten erklären könnte. Die stratigraphischen Übergänge zwischen Jura und Trias sowie zwischen Trias und dem liegenden Perm spiegeln sich im Leitfähigkeitsmodell nicht wider. Vielmehr erscheinen die Sedimente im Liegenden des mittleren Jura bis zum präkambrischen Basement gleichförmig elektrisch gut leitend. Die Basis des sedimentären Schichtpaketes (Kreide, Jura, Trias, Perm) und der Übergang zum präkambrischen Grundgebirge ist im Leitfähigkeitsmodell aufgelöst. Die sedimentäre Schichten bilden eine Wechselfolge guter und schlechter Leiter, hangend auf resistivem Basement. Aufgrund der groben Diskretisierung des 3D Modells in geringen Tiefen ist dort die Wechsellagerung nicht so gut aufgelöst. Aus dem gleichen Grund erscheinen die Oberkanten des präkambrischen Basements an der östlichen und westlichen

Schulter im 3D Modell schlechter aufgelöst als im 2D Modell.

- Über die östliche, jordanische Seite des Untersuchungsgebietes ist nur wenig geologische Hintergrundinformation verfügbar. Dadurch ist die Interpretation des Leitfähigkeitsmodells erschwert. Allerdings konnten hydrologische Information mit der obersten, gut leitenden Schicht im 2D Modell korreliert werden. Demnach handelt es sich dabei um die Belqa-Formation, die von jordanischen Hydrologen und Geologen als regionaler Aquifer betrachtet wird. Ähnlich wie auf der oben diskutierten westlichen Flanke ist diese Schicht im 2D Modell besser aufgelöst als im 3D Modell.
- Die Oberflächenspur der EBF tritt sowohl im 2D Modell als auch im 3D Modell gut aufgelöst als lateraler Kontrast zwischen einem resistiven östlichen Block und einer gut leitenden Zentralstruktur in Erscheinung. Im 3D Modell ist die leitenden Zentralstruktur allerdings undeutlicher ausgeprägt. Andererseits treten im 3D Modell flachliegende, in NS-Richtung lang gestreckte Leiter auf, die den Solen des Toten Meeres und den leitfähigen Beckensedimenten entsprechen. Das 3D Inversionsmodell ebenso wie die 3D Vorwärtsstudien (mit welchen versucht wurde, das Verhalten der Induktionspfeile zu erklären), lieferten für die leitende Zentralstruktur eine Mächtigkeit von ca. 17 km. Insbesondere das 3D Inversionsmodell, das auf der Anpassung der Nebendiagonalelemente des Impedanzensors basiert, deutet mächtigere und besser leitfähige Sedimentfüllungen im südlichen Teilbecken im Vergleich zum nördlichen Teilbecken an. Dieses Ergebnis würde mit [Ben-Avraham & Shubert \(2006\)](#) übereinstimmen, die annehmen, dass das südliche Teilbecken tiefer reicht als das nördliche.
- Unter der Annahme vergleichbarer Strukturen an der WBF und der EBF ist die Oberflächenspur der WBF in den 3D Modellen besser aufgelöst. Im 3D Modell treten an der EBF und der WBF vergleichbare laterale Kontraste auf, die allerdings so nicht im 2D Modell enthalten sind. Im 2D Modell wird eine tiefreichende, subvertikale Struktur abgebildet, die nicht klar von den flacheren leitfähigen Strukturen im Beckenzentrum zu trennen ist, und die von den östlichen und westlichen resistiven Randblöcken begrenzt wird. Untersuchungen zur 2D-Modellauflösung in diesem Bereich zeigen, dass diese Strukturen mindestens bis in die mittlere Kruste reichen.
- Die quartären Beckensedimente korrelieren mit äusserst leitfähigen Strukturen, die im 2D Modell bis in mindestens 3 km Tiefe reichen. Diese Leitfähigkeitsanomalie könnte durch die Solen des Toten Meeres verursacht sein, die mehrere Kilometer tief in die Sedimente migrieren. Genaue Mächtigkeitsangaben zum Becken sind hier allerdings schwierig, da sich diese Oberflächenstruktur mit den tiefen, subvertikalen guten Leitern im Zentrum überlagert und eine Trennung der

Beckensedimente vom Basement im Sinne der elektrischen Leitfähigkeit uneindeutig ist. Wie oben bereits ausgeführt, zeigt das 3D Inversionsmodell, das aus den Nebendiagonalelementen des Impedanztensors abgeleitet wurde, die Existenz zweier leitfähiger Strukturen, die mit dem nördlichen und südlichen Teilbecken übereinstimmen. Allerdings fehlen im 3D Modell im zentralen Beckenbereich unterhalb der Lisan-Halbinsel die im 2D Modell enthaltenen Hinweise auf tiefreichende gut leitfähige Strukturen. Möglicherweise sind die nördlichen und südlichen leitfähigen Teilbecken im 2D Modell als tiefreichende, leitende Struktur im zentralen Bereich invertiert worden, da bei der 2D Inversionsrechnung die Annahme lang ausgedehnter Strukturen in N-S-Richtung gemacht wurde. Hier müssen wir allerdings zwischen der Beckenfüllung und den Beckenranstörungen unterscheiden, da zwei verschiedenen Szenarien aufgrund der Inversionsergebnisse nicht unterschieden werden können: (i) die Randstörungen treten selber als subvertikale Strukturen hoher elektrischer Leitfähigkeit auf, die sich mit den angrenzenden leitenden Beckensedimenten überlagern oder (ii) die Randstörungen sind selber elektrisch nicht leitend, sondern bilden lediglich den Übergang der leitenden Beckensedimente von den resistiven Basementstrukturen ab. Im ersten Fall wären die Abbilder der Randstörungen vergleichbar mit jenen anderer grosser Scherzonen. In der Literatur wird die Tote-Meer-Störung (DST) oft mit der San Andreas Verwerfung (SAF) verglichen, insbesondere wenn die DST auf ihrer gesamten Länge betrachtet wird. Geophysikalische, geologische und geochemische Untersuchungen an der SAF zeigten, dass die Dynamik des Störungssystems und die Prozesse, die zu Erdbeben führen können, eng mit dem (superhydrostatischen) Fluidhaushalt zusammenhängen (Zoback *et al.*, 1987; Becken *et al.*, 2008). Zusätzlich zu meteorischen und Formationswässern, die als ursächlich für den oberflächennahen "fault zone conductor" (FZC, erhöhte Leitfähigkeit in der Umgebung der Verwerfung nahe an der Oberfläche) betrachtet werden, gibt es Hinweise auf die Existenz von Mantelfluiden, die zu den hohen Leitfähigkeiten eines Fluidkanals beitragen, welcher von der Unterkruste oder sogar vom oberen Erdmantel bis in geringe Tiefen reichen kann. Ritter *et al.* (2005) verglichen die elektrischen Signaturen von vier grossen Störungssystemen (einschliesslich der DST, aber an andere Stelle als in dieser Arbeit untersucht). Sie schlugen vor, dass die meisten grossen Störungssysteme durch Zonen erhöhter elektrischer Leitfähigkeit bis in mehrere Kilometer Tiefe (< 10 km) charakterisiert seien. Die hohen Leitfähigkeiten im oberen Teil aktiver Störungen sei auf Fluide in der sogenannten "damage zone" zurückzuführen, einem Bereich, in dem die Gesteine durch Deformation stark geklüftet sind. Allerdings fehlt der FZC an dem Araba Segment der DST; hier scheint die Störung als impermeable Barriere für horizontalen Fluidfluss zu wirken (Ritter *et al.*, 2003 2005). Dieselben Autoren schlugen vor, dass Deformation über einen langen Zeitraum entlang einer

schmalen, wenige Meter breiten damage zone lokalisiert gewesen sein könnte. Angenommen wird hierzu eine lange existente unterschiedlicher Gesteinsfestigkeit von Störungskern und umgebenden Gestein. Offensichtlich ist die Situation am extensionellen pull-apart Becken des Toten Meers ganz anders, da die DST dort in zwei parallel Randstörungen aufspaltet. Ähnlich wie an der DST im Süden finden wir im oberflächennahen Bereich der Störungen keinen FZC. Allerdings können wir deren Existenz nicht völlig ausschliessen, da sie von den ebenfalls hochleitfähigen Beckensedimenten verdeckt sein könnten. Andererseits erinnert das Abbild der tiefreichenden Leitfähigkeitsanomalie unter dem DSB Becken an die Modelle von der SAF nahe Parkfield, wo eine tiefreichende, subvertikale Zone mit hoher elektrischer Leitfähigkeit Pfade für Fluidaufstieg aus dem Modell darstellen könnte.

Zusätzlich zum Abgleich mit den verfügbaren geologischen Vergleichsdaten habe ich ein 2D Modell der seismischen P-Wellengeschwindigkeit (DESIRE seismisches Weitwinkel - Experiment) zum Vergleich für die Interpretation herangezogen. Das seismische Basement stimmt demnach mit der Oberkante der westlichen und östlichen, resistiven Grundgebirgsblöcke aus dem 2D Leitfähigkeitsmodell überein. Allerdings hat das seismische Basement im Beckenbereich im zentralen Profilabschnitt keine Entsprechung im Leitfähigkeitsmodell. Aufgrund der geringen Sensitivitäten der seismischen Weitwinkeldaten gegenüber kleinräumigen, oberflächennahen Strukturen in den oberen 5 km kann in diesem Bereich kein Vergleich durchgeführt werden.

Insgesamt kann man sagen, dass die Ergebnisse der 3D Vorwärtsmodellierung und Inversion der DESIRE-MT Daten einerseits eine gute Korrelation zwischen den 2D und 3D Ergebnissen aufweisen, andererseits aber auch Strukturen enthalten, die mit dem zweidimensionalen Ansatz nicht aufgelöst werden können. Insbesondere möchte ich hier auf eine in EW Richtung ausgedehnte leitfähige Struktur hinweisen, die sich nördlich des westlichen Profilabschnitts (auf jordanischer Seite) befindet. Diese Struktur ist im 3D Modell enthalten, und hat eine Tiefenausdehnung von etwa 3.4 km bis über 20 km. Der südliche Rand dieser Struktur liegt unter der Oberflächenspur der Swiqa-Störung. Allerdings sind hier keine MT Daten gemessen worden, und es fehlen andere geophysikalische und geologische Vergleichsdaten sodass eine Interpretation dieses tiefreichenden Leiters nicht gegeben werden kann.

Karten der Realteil der Induktionspfeile bilden typischerweise Strukturen abseits des Profils ab. Grossräumige, dreidimensionale Modellstudien, die sowohl das Mittelmeer als auch das Rote Meer in die Modellrechnung einschliessen, haben gezeigt, dass die Induktionspfeile bei langen Perioden deutlich durch die Existenz des Mittelmeers beeinflusst sind.

Zusammenfassend zeigt die Arbeit in meiner Dissertation die Vor- und Nachteile von 2D und 3D Inversionsrechnungen auf. Ich bin davon überzeugt, dass die Zukunft in 3D

Inversionen liegt, aber (i) wir benötigen dazu Daten mit guter räumlicher Abdeckung (ii) wir müssen die Ergebnisse von 2D und 3D Inversionen gründlich vergleichen (iii) wir müssen unterschiedliche Inversionstrategien in 3D verstehen und bewerten, ähnlich wie dies für 2D Inversion möglich ist (iv) bessere, effizienter 3D Inversionsalgorithmen müssen zukünftig entwickelt werden.

Chapter 6

Acknowledgements

I am heartily thankful to my supervisor, PD Dr. Oliver Ritter, for his encouragement, supervision, and support from the early to the concluding stages of my dissertation. The constructive discussions with Dr. Ritter did not only allow me to better understand the magnetotelluric method, but also guided me to the right way for this work. The long experience of Dr. Ritter in dealing with MT data helped me greatly to analyze and interpret the DESIRE-MT data.

I gratefully thank Dr. Ute Weckmann. By her encouragement and her long experience in managing and carrying out large field campaigns, we managed to successfully complete the MT measurements in Jordan. Many instructive discussions with Dr. Weckmann about the obtained 2D and 3D models provided alternative views and helped me to develop my final modelling results. I would also like to thank Dr. Weckmann for many comments and corrections of earlier versions of this thesis.

I am also very grateful to Dr. Michael Becken for explaining the theoretical background of MT, which helped me greatly to extend my knowledge about the physical meaning of the MT. Moreover, I would like to acknowledge Dr. Becken's numerous small and quick tips for processing and inverting the DESIRE-MT data.

I would like to thank Dr. Gerard Munoz. Dr. Munoz taught me in my first measuring campaign how to deal with MT instruments and how to install an MT station. Dr. Munoz and Dr. Ritter also supervised the MT data acquisition in Israel.

I would like to thank Dr. Ritter for the opportunity to participate in national and international conferences where I met Prof. Dr. G. Egbert. In the course of my PhD work I was then able to work with Prof. Egbert at Oregon State University (Corvallis, USA) for three months. I gratefully acknowledge his hospitality and his introduction to the world of 3D MT inversion. In this respect I would also like to thank also Dr. Anna Kelbert for many useful discussions during my stay in Corvallis.

I wish to sincerely thank all my colleagues at the GFZ; the collaboration with them

was of utmost importance for the preparation of my thesis. They shared their knowledge and made many helpful suggestions. In particular, I would like to thank my office-mate Dipl. Geophysicist Kristina Tietze for discussions about the static shift and perhaps more importantly for her patience when having to cope with me under stress but also for German translations. Also I would like to thank my other office-mate, Dipl. Geophysicist Xiaoming Chen with whom I enjoyed discussions about 2D modelling of (anisotropic) MT data. Dr. Rita Streich also read parts of this thesis and shared my interest in parallel computing. Many thanks go to all other members of the MT-team at the GFZ and to the colleagues from the Free University for our many fruitful discussions in our MT-AG meetings.

Above all, I wish to thank my wife Carmen, who has witnessed the development of this thesis step by step, not only on the computer screen, but also over a long time span with her nerves. Without Carmen's understanding, patience, and encouragement I could not have done this work. While I am deeply thankful for all of this, I regret to have lost many nice moments and days with you Carmen because I was so busy. I would also like to very sincerely thank my mother, who would have loved to see her son become a doctor. Great thanks go to my sisters and brothers who supported me at all stages of my life.

I would like to extend my heartfelt gratitude to all people who helped collecting MT data in Jordan and Israel:

Michal Becken, Jana Börner, Romina Gehrmann, Olaf Helwig, Juliane Hübner, Thomas Krings, Christian Mielke, Gerard Munoz, Stefanie Musiol, Stefan Rettig, Oliver Ritter, Manfred Schüler, Ute Weckmann and Wenke Wilhelms (Germany). Khalil Abu-Ayyash, Khaldon Abu Hamideh, Darwish Jaser, Jamal Khataibeh, Issam Qabbani, Husam Al-Rashdan and Tahsin Tal'at (Jordan). Yossi Bartov, Uri Frieslander and Gabby Heim (Israel).

Finally, I would like to thank the the German Research Foundation (Deutsche Forschungsgemeinschaft, (DFG) for the financial support, Prof. M. Weber for his great organisation of the DESIRE project, and the Geophysical Instrument Pool-Potsdam (GIPP) for providing the MT instruments.

Bibliography

- Al-Zoubi, A., & Ben Avraham, Z. 2002. Structure of the earth's crust in Jordan from potential field data. *Tectonophysics*, 346(1-2), 45–59. [cited at p. 96]
- Al-Zoubi, A., & Ten Brink, U.S. 2001. Salt diapirs in the Dead Sea basin and their relationship to Quaternary extensional tectonics. *Marine and Petroleum Geology*, 18, 779–797. [cited at p. 95, 98, 99, 100]
- Al-Zoubi, Abdallah, & Brink, Uri S. 2001. Salt diapirs in the Dead Sea basin and their relationship to Quaternary extensional tectonics. *Marine and Petroleum Geology*, 18(7), 779–797. [cited at p. 173, 181]
- Alumbaugh, D.L., Newman, G.A., L., Prevost, & Shadid, J.N. 1996. Three-dimensional wideband electromagnetic modeling on massively parallel computers. *Radio Science*, 31, 1–23. [cited at p. 21, 23, 35, 58, 61, 62]
- Bahr, K. 1988. Interpretation of the magnetotelluric impedance tensor: regional induction and local distortion. *J. Geophys*, 62, 119127. [cited at p. 108, 111]
- Bahr, Karsten. 1991. Geological noise in magnetotelluric data: a classification of distortion types. *Physics of The Earth and Planetary Interiors*, 66(1-2), 24 – 38. [cited at p. 111]
- Barjous, M., & Mikbel, Sh. 1990. Tectonic evolution of the gulf of Aqaba-Dead Sea transform fault system. *Tectonophysics*, 180(1), 49 – 59. Geologic and Tectonic Processes of the Dead Sea Rift Zone. [cited at p. 93]
- Becken, M., & Burkhardt, H. 2004. An ellipticity criterion in magnetotelluric tensor analysis. *Geophysical Journal International*, 159(1), 69–82. [cited at p. 108, 109]
- Becken, M., Ritter, O., Park, S. K., Bedrosian, P. A., Weckmann, U., & Weber, M. 2008. A deep crustal fluid channel into the San Andreas Fault system near Parkfield, California. *Geophysical Journal International*, 173(2), 718–732. [cited at p. 175, 183]
- Ben-Avraham, Z., & Shubert, G. 2006. Deep drop down basin in the southern Dead Sea. *Earth and Planetary Science Letters*, 251, 254–263. [cited at p. 95, 96, 97, 158, 167, 174, 182]

- Ben-Avraham, Zvi, & Grasso, Mario. 1991. Crustal structure variations and transcurrent faulting at the eastern and western margins of the eastern Mediterranean. *Tectonophysics*, 196(3-4), 269–277. [cited at p. 99]
- Ben-Avraham, Zvi, & Ten Brink, Uri. 1989. Transverse faults and segmentation of basins within the Dead Sea Rift. *Journal of African Earth Sciences*, 8(2-4), 603 – 616. [cited at p. 95]
- Ben-Avraham, Zvi, Garfunkel, Zvi, & Lazar, Michael. 2008. Geology and Evolution of the Southern Dead Sea Fault with Emphasis on Subsurface Structure. *Annual Review of Earth and Planetary Sciences*, 36(1), 357–387. [cited at p. 93]
- Bruce, R. A. A., Macdonald, N. B., & Trew, A. S. 1993. *CHIMP and PUL: Support for Portable Parallel Computing*. Tech. rept. In Proceedings of the Fourth Annual Conference of the Meiko User Society. [cited at p. 46]
- Cagniard, L. 1953. Basic Theory of the Magneto-Telluric Method of Geophysical Prospecting. *Geophysics*, 18, 605–635. [cited at p. 7, 10, 11]
- Commer, Michael, & Newman, Gregory A. 2009. Three-dimensional controlled-source electromagnetic and magnetotelluric joint inversion. *Geophysical Journal International*, 178(3), 1305–1316. [cited at p. 62]
- Constable, S. C., L., Parke R., & G., Constable C. 1987. Occams inversion - A practical algorithm for generating smooth models from electromagnetic sounding data. *Geophysics*, 52(3), 289300. [cited at p. 26, 29, 30]
- deGroot Hedlin, C., & Constable, S. 1990. Occams inversion to generate smooth, two-dimensional models from magnetotelluric data. *Geophysics*, 55(12), 1613–1624. [cited at p. 30]
- Egbert, G. 1997. Robust multiple station magnetotelluric data processing. *Geophysical Journal International*, 130, 475–496. [cited at p. 103]
- Egbert, Gary D. 2006. *Efficient Inversion of Multi-frequency and Multi-Source Electromagnetic Data*. Final Project Report. College of Oceanic and Atmospheric Sciences, Oregon State University, College of Oceanic and Atmospheric Sciences, Oregon State University. [cited at p. 20, 21, 26, 30, 31, 33, 34, 39, 40, 74, 78]
- Favetto, Alicia, Pomposiello, Cristina, de Luchi, Monica G. Lopez, & Booker, John. 2008. 2D Magnetotelluric interpretation of the crust electrical resistivity across the Pampean terrane Rio de la Plata suture, in central Argentina. *Tectonophysics*, 259, 5465. [cited at p. 121]
- Flynn, M. 1972. Some Computer Organizations and Their Effectiveness. *IEEE Trans. Comput.*, C-21, 948+. [cited at p. 44, 45]
- Forum, MPI. 2008. *MPI: a message-passing interface standard: Version 2.1 ; June 23, 2008*. Knoxville, Tenn.: Univ. of Tennessee. [cited at p. 48]

- Gamble, T. D., Goubau, W. M., & Clarke, J. 1979. Magnetotellurics with a remote magnetic reference. *Geophysics*, 44(1), 53–68. [cited at p. 103]
- Garfunkel, Z., Zak, I., & Freund, R. 1981. Active faulting in the dead sea rift. *Tectonophysics*, 80(1-4), 1 – 26. The Dead Sea Rift. [cited at p. 92, 93, 95, 96]
- Garfunkel, Zvi. 1981. Internal structure of the Dead Sea leaky transform (rift) in relation to plate kinematics. *Tectonophysics*, 80(1-4), 81–108. The Dead Sea Rift. [cited at p. 93, 95, 96]
- Garfunkel, Zvi, & Ben-Avraham, Zvi. 1996. The structure of the Dead Sea basin. *Tectonophysics*, 266, 155–176. [cited at p. 142, 143]
- Ginzburg, A., & Ben-Avraham, Z. 1997. A seismic refraction study of the northern basin of the Dead Sea, Israel. *Geophys. Res. Lett.*, 24, 20632066. [cited at p. 96]
- Hofstetter, Rami, Klinger, Yann, Amrat, Abdel-Qader, Rivera, Luis, & Dorbath, Louis. 2007. Stress tensor and focal mechanisms along the Dead Sea fault and related structural elements based on seismological data. *Tectonophysics*, 429(3-4), 165–181. [cited at p. 95]
- Jackson, D. D. 1972. Interpretation of Inaccurate, Insufficient and Inconsistent Data. *Geophysical Journal of the Royal Astronomical Society*, 82(2), 97–109. [cited at p. 26]
- Jiracek, G. R. 1990. Near-surface and topographic distortion in electromagnetic induction. *Surveys in Geophysics*, 11, 163–203. [cited at p. 7, 8, 115]
- Kafri, U., Goldman, M., & Levi, E. 2008. The relationship between saline groundwater within the Arava Rift Valley in Israel and the present and ancient base levels as detected by deep geoelectromagnetic soundings. *Environ. Geol.*, 54, 14351445. [cited at p. 144]
- Kashai, E.L., & Croker, P.F. 1987. Structural geometry and evolution of the Dead Sea - Jordan rift system as deduced from new subsurface data. *Tectonophysics*, 141, 33–60. [cited at p. 144, 145]
- Keller, V. G. 1988. Physical Laws, Electromagnetic Induction and Rock Properties. *Chap. 2, pages 13–51 of*: Nabighian, Misac N. (ed), *Electromagnetic Methods in Applied Geophysics*. Society of Exploration Geophysics (SEG). [cited at p. 9, 10]
- Khalil, B. 1992. *The geology of the Ar Rabba area*. Geological Map Bulletin No. 22. Natural Resource Authority (Jordan), Amman, Jordan. [cited at p. 96, 97, 144, 146]
- Klinger, Y., Avouac, J. P., Karaki, N. Abou, Dorbath, L., Bourles, D., & Reyss, J. L. 2000. Slip rate on the Dead Sea transform fault in northern Araba valley (Jordan). *Geophysical Journal International*, 142(3), 755–768. [cited at p. 93]
- Kowalik, Janusz S., & Grandinetti, Lucio (eds). 1993. *Software for Parallel Computation: Proceedings of the NATO Advanced Research Workshop on Software for Parallel Computation, Held at Cetraro, Cosenza, Italy, June 22-26, 1992*. Secaucus, NJ, USA: Springer-Verlag New York, Inc. [cited at p. 46]

- Krings, T. 2007. *The influence of Robust Statistics, Remote Reference, and Horizontal Magnetic Transfer Functions on data processing in Magnetotellurics*. M.Phil. thesis, Institut fuer Geophysik stfaealische Wilhelms-Universitaet Muenster and GeoForschungsZentrum Potsdam. [cited at p. 102]
- Le Beon, M., Klinger, Yann., Abdel Qader, Amrat, , Amotz, Agnon, Louis, Dorbath, Gidon, Baer, Jean-Claude, Ruegg, Olivier, Charade, & Omar, Mayyas. 2008. Slip rate and locking depth from GPS profiles across the southern Dead Sea Transform. *J. Geophys. Res.*, 113. [cited at p. 93]
- Levenberg, K. 1944. A Method for the Solution of Certain Problems in Least Squares. *Quart. Appl. Math.*, 2, 164-168. [cited at p. 29]
- Mackie, R. 1996. Mathematical Methods for Geo-Electromagnetic Induction. *Physics of the earth and planetary interiors*, Vol. 97, No. 1/4 (1996), 279–280. [cited at p. 23]
- Mackie, R. L., Smith, J. T., & ., T. R Madden. 1994. Three-dimensional electromagnetic modeling using finite difference equations: The magnetotelluric example. *Radio Sci.*, 29(4), 923-935. [cited at p. 22]
- Madsen, K., Nielsen, H.B., & Tingleff, O. 2004. *METHODS FOR NON-LINEAR LEAST SQUARES PROBLEMS*. Tech. rept. Informatics and Mathematical Modelling Technical University of Denmark. [cited at p. 29]
- Masri, A. 2003. *The geology of Dhiban (Wadi Al Mujib) area*. Geological Map Bulletin No. 54. Natural Resource Authority (Jordan), Amman, Jordan. [cited at p. 96]
- Mechie, J., Abu-Ayyash, K., Ben-Avraham, Z., El-Kelani, R., Qabbani, I., Weber, M., & Team, DESIRE. 2009. Crustal structure of the southern Dead Sea basin derived from project DESIRE wide-angle seismic data. *Geophysical Journal International*, 178(1), 457–478. [cited at p. 99, 146, 147]
- Meju, M.A. 1994. *Geophysical Data Analysis: Understanding Inverse Problem Theory and Practice*. 6. Tulsa, Oklahoma: Society of Exploration Geophysicists. [cited at p. 31]
- Menke, William. 1984. *Geophysical data analysis : discrete inverse theory*. Academic Press, Orlando, Fla. [cited at p. 31]
- Neev, David, & Hall, John K. 1979. Geophysical investigations in the Dead Sea. *Sedimentary Geology*, 23(1-4), 209 – 238. [cited at p. 95, 96]
- Newman, G. A., & Alumbaugh, D. L. 1997. Three-dimensional massively parallel electromagnetic inversion I. Theory. *Geophysical Journal International*, 128(2), 345–354. [cited at p. 21, 61]
- Newman, G. A., & Alumbaugh, D. L. 2000. Three-dimensional magnetotelluric inversion using non-linear conjugate gradients. *Geophysical Journal International*, 140, 410–424. [cited at p. 34, 38, 61]

- Porstendorfer, Gottfried. 1975. *Principles of magneto-telluric prospecting*. Geoexploration monographs. Series 1 ; no. 5. Berlin ; Stuttgart : Borntraeger. [cited at p. 11, 13]
- Powell, John H. 1988. The geology of the Karak area map sheet No. 3152III. *Geology Directorate, Natural Resources Authority, Amman-Jordan*. [cited at p. 97, 142, 146]
- Ritter, O., Junge, A., & Dawes, G. 1998. New equipment and processing for magnetotelluric remote reference observations. *Geophysical Journal International*, 132, 535–548. [cited at p. 102]
- Ritter, O., Ryberg, T., Weckmann, U., Hoffmann-Rothe, A., Abueladas, A., Z.Garfunkel, & Group, DESERT. 2003. Geophysical images of the Dead Sea Transform in Jordan reveal an impermeable barrier for fluid flow. *Geophysical Research Letters*, 30(14), 1741–1744. [cited at p. 175, 183]
- Ritter, O., Hoffmann-Rothe, A., Bedrosian, P. A., Weckmann, U., & Haak, V. 2005. *Electrical conductivity images of active and fossil fault zones*. Vol. 245. The Geological Society, London. Chap. High Strain Zones: Structure and Physical Properties. [cited at p. 175, 183]
- Rodi, W., & Mackie, R. L. 2001. Nonlinear conjugate gradients algorithm for 2-D magnetotelluric inversion. *Geophysics*, 66(1), 174–187. [cited at p. 30, 31, 34, 38, 39, 117, 118, 171, 179]
- Rodi, W. L. 1976. A technique for improving the accuracy of finite element solutions for magnetotelluric data. *Geophys. J. Roy. Astr. Soc.*, 44, 483506. [cited at p. 31, 34]
- Simpson, Fiona, & Bahr, Karsten. 2005. *Practical magnetotellurics*. Cambridge UK , New York: Cambridge University Press. [cited at p. 125]
- Siripunvaraporn, W., & Egbert, G. 2000. An efficient data-subspace inversion method for 2-D magnetotelluric data. *Geophysics*, 65(3), 791–803. [cited at p. 40, 74]
- Siripunvaraporn, W., Egbert, G., & Lenbury, Y. 2002. Numerical accuracy of magnetotelluric modeling: A comparison of finite difference approximations. *LETTER Earth Planets Space*, 54, 721–725. [cited at p. 22]
- Siripunvaraporn, W., Egbert, G., Lenbury, Y., & Uyeshima, M. 2005. Three-dimensional magnetotelluric inversion: data-space method. *Physics of the Earth and Planetary Interiors*, 150(May), 3–14. [cited at p. 21, 34, 40, 41, 61, 74, 78]
- Siripunvaraporn, Weerachai, & Egbert, Gary. 2009. WSINV3DMT: Vertical magnetic field transfer function inversion and parallel implementation. *Physics of the Earth and Planetary Interiors*, 173(3-4), 317 – 329. [cited at p. 61, 62]
- Smith, Torquil J. 1996a. Conservative modeling of 3-D electromagnetic fields, Part I: Properties and error analysis. *Geophysics*, 61(5), 1308–1318. [cited at p. 19, 23]
- Smith, Torquil J. 1996b. Conservative modeling of 3-D electromagnetic fields, Part II: Bi-conjugate gradient solution as an accelerator. *Geophysics*, 61(5), 1319–1324. [cited at p. 22]

- Telford, W. M. and Geldart, L. P., & Sheriff, R. E. 1990. *Applied geophysics (2nd ed.)*. Cambridge, Cambridge University Press. [cited at p. 142]
- Ten Brink, U.S., Ben-Avraham, Z., Bell, R. E., Hassouneh, M., Coleman, D. F., Andreasen, G., Tibor, G., , & Coakley, B. 1993. Structure of the Dead Sea Pull-Apart Basin From Gravity Analyses. *Journal of Geophysical Research*, 98(B12), 21,877–21,894,. [cited at p. 96]
- Vozoff, Keeva. 1972. THE MAGNETOTELLURIC METHOD IN THE EXPLORATION OF SEDIMENTARY BASINS. *Geophysics*, 37(1), 98–141. [cited at p. 10]
- Wait, J.R. 1954. On the relation between telluric currents and the Earth's magnetic field. *Geophysics*, 19, 281–289. [cited at p. 13]
- Ward, H. Stanley, & Hohmann Gerald W. 1987. Electromagnetic Theory for Geophysical Applications. *Pages 131–308 of: M. N. Nabighian (ed), Electromagnetic Methods in Applied Geophysics. Volume 2, Application, Parts A and B*. SEG, Tulsa. [cited at p. 6]
- Weaver, J. T. 1994. *Mathematical Methods for Geo-electromagnetic Induction*. Applied and Engineering Mathematic Series. Taunton, Somerset, Enagland: Research Studies Press. [cited at p. 7, 9, 10, 13, 14, 19, 21, 23, 34, 35, 106, 117, 125, 131]
- Weckmann, U., Magunia, A., & Ritter, O. 2005. Effective noise separation for magnetotelluric single site data processing using a frequency domain selection scheme. *Geophysical Journal International*, 161, 635–652. [cited at p. 102]
- Weckmann, Ute, Ritter, Oliver, & Haak, Volker. 2003. Images of the magnetotelluric apparent resistivity tensor. *Geophysical Journal International*, 155(2), 456–468. [cited at p. 10]
- Wiese, H. 1962. Geomagnetische Tiefentellurik Teil: II. die Streichrichtung der Untergrundstrukturen des elektrischen Widerstandes, erschlossen aus geomagnetischen Variationen. *Geofis. Pura Appl.*, 52, 83103. [cited at p. 10, 106]
- Yee, K. S. 1966. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. *IEEE Trans. Antennas Propag.*, 14, 302307. [cited at p. 21]
- Zak, I., & Freund, R. 1981. Asymmetry and basin migration in the dead sea rift. *Tectonophysics*, 80(1-4), 27–38. The Dead Sea Rift. [cited at p. 92]
- Zoback, Mark D., Zoback, Mary Lou, Mount, Van S., Suppe, John, Eaton, Jerry P., Healy, John H., Oppenheimer, David, Reasenberg, Paul, Jones, Lucile, Raleigh, C. Barry, Wong, Ivan G., Scotte, Oona, & Wentworth, Carl. 1987. New Evidence on the State of Stress of the San Andreas Fault System. *Science*, 238(4830), 1105–1111. [cited at p. 175, 183]

Appendices

Appendix A

Appendix

To better understand the meaning of the previous mathematical formulation, let us consider the following example in 2D. Given is a data vector (\mathbf{d}) which contains the impedances Z_{xy} (E-polarisation) for 5 periods at 3 stations ($N=5 \times 3 \times 2=30$)¹. The induction forward modelling problem ($\mathbf{A}\mathbf{e} = \mathbf{b}$) is solved for the electric field \mathbf{e} defined on $N_y \times N_z$ nodes² of the 2D grid, where N_y and N_z are the number of nodes in y and z direction, respectively. Since we have only Z_{xy} as measured data, the computed model responses for period Per_i at station Stn_j is:

$$F^{Per_i, Stn_j}(\mathbf{m}) = Z_{xy}^{Per_i, Stn_j} = \frac{E_x^{Per_i, Stn_j}}{B_y^{Per_i, Stn_j}} = \frac{\lambda_E \mathbf{e}}{\lambda_B \mathbf{e}} = \psi(\mathbf{e}(\mathbf{m}), \mathbf{m})$$

The basic functions λ_E and λ_B interpolate the electric field at one particular station and period to evaluate E_x and B_y . Suppose that we want to know how sensitive the 30 predicted data points are towards a small perturbation of the first model parameter (m_1). This means that we have to compute the values in the first column of \mathbf{J} :

$$\mathbf{J}_{l,1} = \frac{\partial \mathbf{F}_l(\mathbf{m})}{\partial m_1} \quad l = 1, \dots, 30$$

To compute these values of \mathbf{J} using eq. 2.80, first we must solve $\mathbf{A}^{-1}\mathbf{L}^T$ and then substitute the result in eq. 2.80. However, solving $\mathbf{A}^{-1}\mathbf{L}^T$ requires computing the columns of \mathbf{L}^T (rows of \mathbf{L}). From the definition of \mathbf{L} we can obtain:

$$\mathbf{L} = \frac{\partial \psi}{\partial \mathbf{e}} = \frac{\partial}{\partial \mathbf{e}} \left(\frac{\lambda_E \mathbf{e}}{\lambda_B \mathbf{e}} \right) \quad (\text{A.1})$$

and using the quotient rule, we obtain the j th row of \mathbf{L} , which corresponds to one particular station and period,

$$L_j = (\lambda_B \mathbf{e})^{-1} \lambda_E - [\lambda_E \mathbf{e} / (\lambda_B \mathbf{e})^2] \lambda_B \quad (\text{A.2})$$

¹Two is to allow for real and imaginary parts.

²Refers to a cell corner in a 2D grid and a cube corner in 3D grid.

The length of L_j is equal to $Ny \times Nz$. However, only a few elements surrounding the station under consideration are non-zero. Repeating the same procedure for all stations and periods gives the full $N \times M$ \mathbf{L} matrix. Taking the transpose of \mathbf{L} , solving $\mathbf{A}^{-1}\mathbf{L}^T$ for one column in \mathbf{L}^T (one station and one period) and then multiplying with $\frac{\partial \mathbf{A}}{\partial m_1} \mathbf{e}$ results in the required element $\frac{\partial F^{Per_i, Stn_j}(\mathbf{m})}{\partial m_1}$ of \mathbf{J} . For the E-polarisation, the function ψ is independent of \mathbf{m} and therefore $\mathbf{Q}=0$.

Figure A.1 demonstrates how \mathbf{J} (Fig. A.1a) or \mathbf{J}^T (Fig. A.1b) are computed for our example. The square sparse coefficients matrix \mathbf{A} contains 42 rows and 42 columns, since we have 42 cell corners ($Ny=6, Nz=5; Ny+1 \times Nz+1 =$ cell corners). The derivative of \mathbf{A} with respect to m_1 , however, contains only 4 non-zero rows corresponding to the cell corners 1,2,7 and 8, because the computation of the gradient and divergences of the EM fields for these cell corners depend on m_1 . Multiplying $\frac{\partial \mathbf{A}}{\partial m_1}$ with \mathbf{e} results in a vector with 42 zero elements except the elements 1,2,7 and 8³. This vector represents one column in \mathbf{P} (i.e. \mathbf{p}_1 in Fig. A.1a). To compute the required sensitivity value at station Stn_1 for the first period with respect to m_1 using eq. 2.79 (Fig. A.1a) we need the first row of \mathbf{L} (\mathbf{l}_1). Computing \mathbf{l}_1 requires the electric field values e_{19}, e_{20}, e_{25} and e_{26} which surround the station location Stn_1 , to evaluate E_x and B_y . The length of \mathbf{l}_1 is also 42 and all elements except for 19, 20, 25 and 26 are zero. Solving $\mathbf{A}^{-1}\mathbf{p}_1$ and multiplying the result with \mathbf{l}_1 gives the required sensitivity value for one particular period. Computing \mathbf{l}_2 and multiplying with $\mathbf{A}^{-1}\mathbf{p}_1$ gives the sensitivity values for one period at station Stn_2 , etc. To compute all values of \mathbf{J} , the same procedure is repeated for all columns of \mathbf{P} ($\mathbf{p}_1, \dots, \mathbf{p}_{30}$) and for all periods. This means that $\mathbf{A}^{-1}\mathbf{p}$ must be solved 150 (30×5) times.

In figure A.1a we set the sources in each model parameter to compute the columns of \mathbf{P} and then solve the pseudo forward modelling problem. However, by setting the sources at each station location Stn_1 (A.1b) and solving the pseudo forward modelling problem $\mathbf{A}^{-1}\mathbf{L}^T$ (reciprocity principle) reduces the number of pseudo forward modelling steps to N (number of columns of \mathbf{L}^T). The same procedure is followed for computing the columns of \mathbf{L}^T and \mathbf{P}^T . However, we first compute \mathbf{l}_1^T , solve $\mathbf{A}^{-1}\mathbf{l}_1^T$ and then multiply the result with all rows of \mathbf{P}^T ($\mathbf{p}_1^T, \dots, \mathbf{p}_{30}^T$ in this example). Eventually, we obtain one row of \mathbf{J} (one column in \mathbf{J}^T).

³The electric field vector \mathbf{e} contains the solution for one period only.

Mein Lebenslauf wird aus Grnden des Datenschutzes in der elektronischen Fassung meiner Arbeit nicht veröffentlicht.